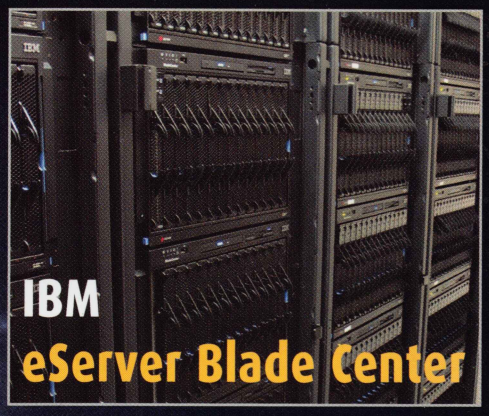
AN SSC PUBLICATION

Build a personal video recorder

LINUX JOURNAL

High-density servers:



The Monthly Magazine of the Linux Community • JANUARY 2004

Astronomy and Telescope Control

RCU for large SMP systems

Signing kernel modules

New solution for air traffic control

Monitor drive health with SMART

Secure mail servers

USA $5.00 CAN $6.50
www.linuxjournal.com
01
0 71486 03102 4

New RAIDZONE *WireSpeed*

Unmatched Performance in All Network Storage Architectures --- NAS, SAN, iSCSI

New Products

Gangstor: Replicated Storage Made Easy

With disk capacities doubling and prices falling, you can buy two RAIDZONE servers for last year's price of one. The importance of data replication and business continuation makes that a very good idea. Our new Gangstor replicated servers make it a no-brainer.

Heard About SAN & NAS Converging? With AllStor, They've Mated

RAIDZONE AllStor combines NAS (NFS, WIndows), iSCSI, and Fibre Channel SAN in one easy-to-configure server appliance. Just create partitions, and allocate them individually to any storage protocol.

OpenNAS for SAN: The Ultimate Front End NAS

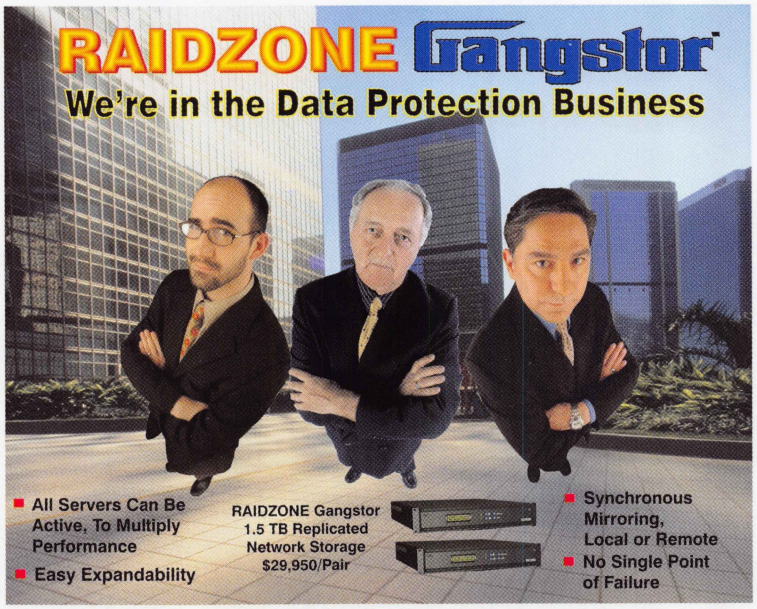
You've invested in a high end SAN. Now get the fastest, most reliable NAS front end. With up to 3 TB of redundant local storage as a cache/mirror of your allocated SAN storage, RAIDZONE OpenNAS for SAN effortlessly serves all your NFS and SMB (Windows) clients and provides transparent replication of your data.

For more information, please visit our Web site:

www.raidzone.com

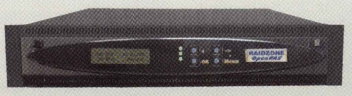
or give us a call at:

1-800-388-1896



The *WireSpeed* RAIDZONE OpenNAS

NFS and Windows (SMB) At Over 100 MB/Sec.



RS8 2U 1.4 TB $14,450

RS15 4U 3 TB $24,950

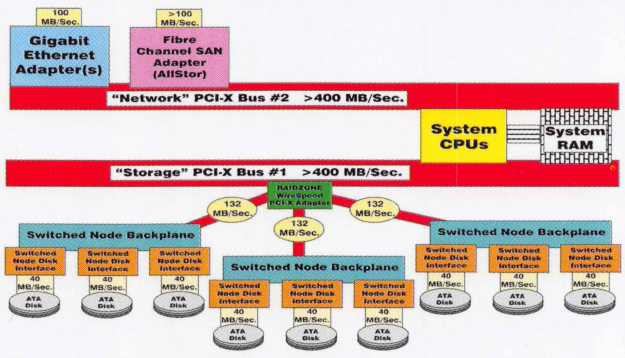
The *WireSpeed* RAIDZONE AllStor

SAN & NAS in One Server: iSCSI, Fibre Channel, NFS, SMB

RA15 4U 3 TB $26,950

RA8 2U 1.4 TB $16,450

RAIDZONE *WireSpeed* Server Architecture



New Ideas

Get a Free 21-Day Eval

Our network storage products sell themselves. If you've got an immediate requirement, we'll send out one of our best salesmen --- AllStor, Gangstor, or OpenNas for San --- to win your business.

RAIDZONE WireSpeed Data Path: Two 500 MB/Sec. PCI-X Buses

Our RAIDZONE disk array technology was the first to connect every disk independently to a 133 MB/Sec. PCI system bus. The new RAIDZONE WireSpeed technology uses one 500 MB/Sec. PCI-X system bus for the disk array, and a separate 500 MB/Sec. PCI-X bus for the network interface.

RAIDZONE WireSpeed Servers Are Serial ATA-Ready

We pioneered the use of Ultra ATA disk arrays in enterprise class storage. Since mid-2000, hundreds of thousands of RAIDZONE NAS users have reaped the benefits in bigger, faster network storage at much lower prices. Today, on the verge of a new generation of ATA drives, we're still leading. Our WireSpeed Servers can use both Ultra ATA and Serial ATA drives.

RAIDZONE is a registered trademark of Consensys Corp.; OpenNAS is a registered trademark and Gangstor, AllStor are trademarks of Consensys Computers Inc.; Windows/Microsoft Corp.; Linux/Linus Torvalds. The image of any identifiable person in this advertisement is used for illustrative purposes only; no endorsement is implied.

ANGSTROM
Microsystems

AMD Opteron™ Specialists

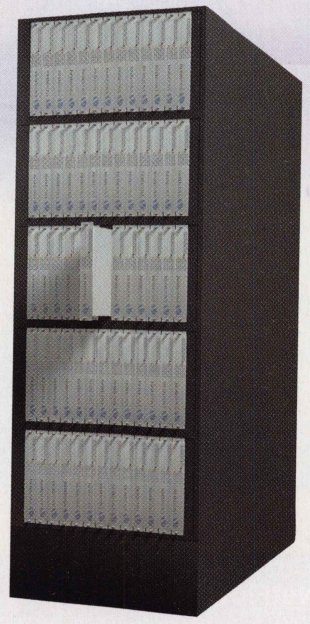
Largest AMD Opteron™ cluster for National Institute of Health installed at the University of Utah.

Over 1,000 CPUs!

AMD Opteron™

World's Densest Server Rack:

Titan64 SuperBlade™
130 SERVER BLADES - 260 AMD Opteron™
and AMD Athlon™ MP Processors in ***One Rack***!



As clusters become the standard in supercomputing solutions, more servers are required to number crunch larger and larger datasets. Titan64 SuperBlades™ solve key problems in the growth of these clusters in the following fashion:

Increased density: 128 server nodes-plus 2 management nodes per rack

Easy to maintain: Servers are hot-swappable and easy to remove

Easy to manage: System administrator can remotely manage multiple racks simultaneously

Integrate with existing datacenters: Customer specified power connectors, front-to-back cooling

Flexible design: Option to reduce density of blades and increase 19" rack space

Improve reliability: Large clusters exhibit problems not found in those that are small

Titan64 SuperBlades™ solution:
- Reduce fan count, use large fans (no small fans or blowers),
- Eliminate most cables, and provide a select indicator for blades to be serviced,
- LinuxBIOS enabled!

Titan64 SuperBlades™ more cost-effective than 1U servers

Titan64 Creator™ Workstations

High Performance Computing

The **Titan64 Creator**™ is a series of high-performance workstations with advanced 8x AGP graphics capabilities integrated with the powerful AMD Opteron™ platform to provide a content creation engine for animation, 3D modeling, and CAD.

Cutting-Edge Performance: 8x AGP graphics performance on Microsoft® and Linux® platforms running Maya®, XSI® and many other graphics intensive creation applications.

Ideal for NVIDIA® Quadro® FX 1000 by PNY®

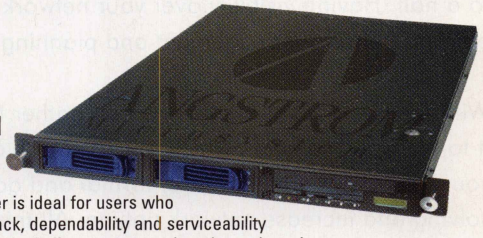
Ultra-Cool Design: Chassis design based on cooling, functionality and ease of access without sacrificing the looks

Focus On Content Creation: Our workstations and servers have created and rendered movies such as X-Men 2™, DAREDEVIL™, Scooby Doo™ The Movie and MIB II™

Proven Quality: Our proven track record for exceptional quality has enabled our clients to deploy mission-critical systems world-wide

* We offer the best price performance in the industry.

AMD Opteron™- based Enterprise Solutions



The **Titan64 Enterprise** server is ideal for users who demand high CPU density per rack, dependability and serviceability for a broad range of applications including storage, web and portal serving, application development, compute farms, and security. Designed for mission-critical 24 x 7 x 365 environments, the Titan64 Enterprise™ delivers on the promises of 64-bit technology for the enterprise today.

The LinuxBIOS Revolution – ideal for blade solutions

:: Fast boot time (less than 5 seconds)

:: Improve quality and reduce cost by eliminating HDD

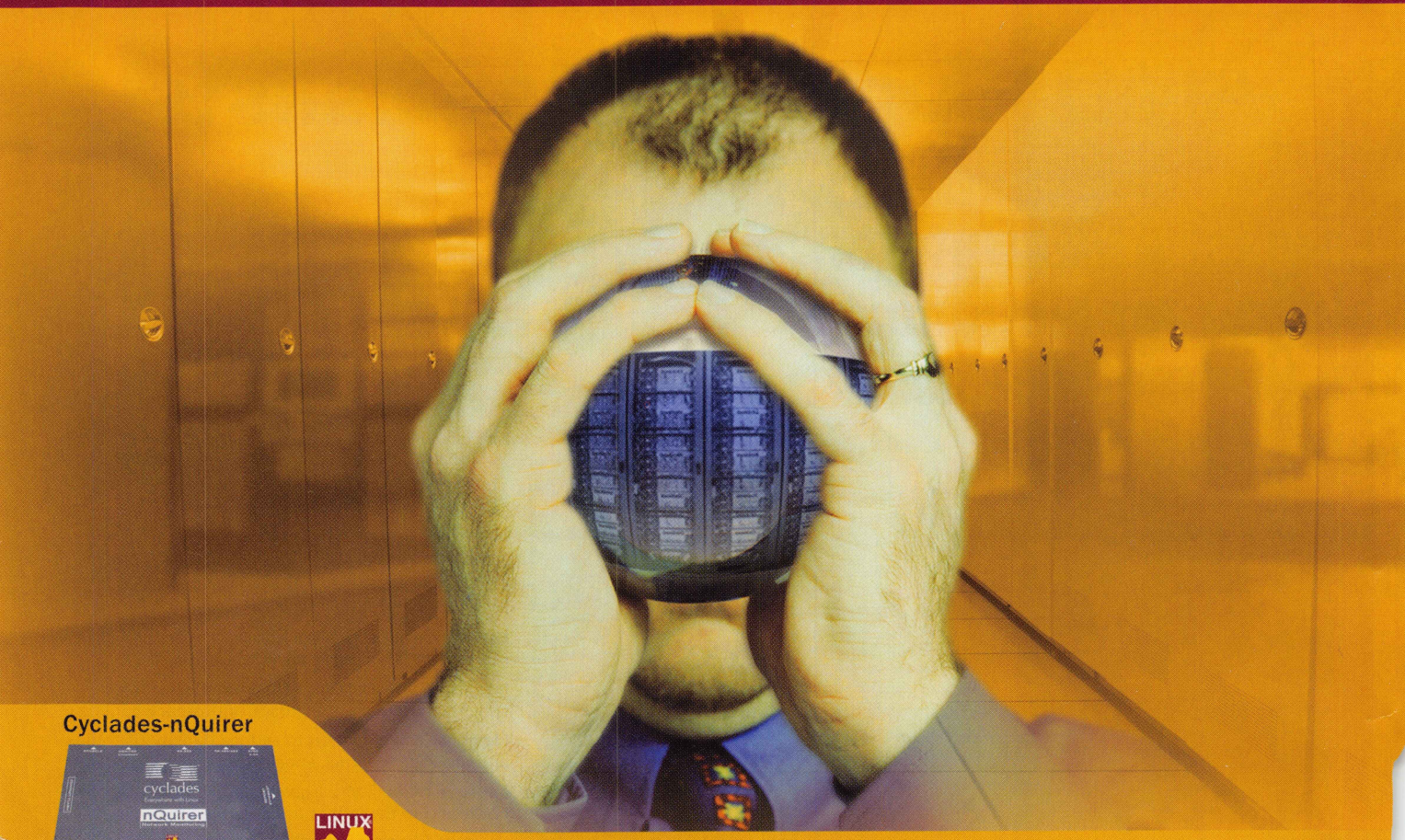
:: Remote server management
- BIOS updates • Console access • Environmental monitoring
- Debugging failed server

:: Easy to patch O/S and upgrade server software

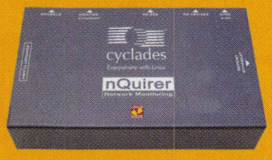
ANGSTROM
Microsystems

Angstrom Microsystems, Inc. • 27 Drydock Ave • Boston, MA 02210
• 1-877-4-ANGSTROM • www.angstrom.com • sales@angstrom.com

Get a closer look inside your network



Cyclades-nQuirer



LINUX INSIDE

Introducing the Cyclades-nQuirer with NetFlow™ support

The complex networks of today carry all different types of traffic, from customer information to viruses that can bring your network to a halt. Having visibility over your network traffic is essential for efficient network management and planning.

With the Cyclades-nQuirer, you can gather traffic data and convert it to NetFlow™ format, which allows you to consolidate and analyze your network in a higher-level format and achieve faster problem solving and increase network uptime. All this in compact form factor and at a fraction of the cost of comparable solutions.

With the Cyclades-nQuirer you will be able to look inside your network and understand it like you've never had before.

LINUX JOURNAL READERS' CHOICE 2003
"Best Hardware for Linux since 1995"

www.cyclades.com/ljb
1.888.cyclades ▪ sales@cyclades.com

cyclades
Everywhere with Linux

©2004 Cyclades Corporation. All rights reserved. All other trademarks and product images are property of their respective owners. Product information subject to change without notice.

LINUX JOURNAL

JANUARY 2004 ISSUE 117

COVER STORY

69 TALON: GPL SOFTWARE FOR TELESCOPE CONTROL

The open-source revolution in astronomy is largely thanks to Talon, the GPL package that handles tracking, scheduled operations, environmental monitoring, dome control and even image analysis. Our author Tony Steidler-Dennison has set up Talon-based telescopes at staffed and automated locations around the world.

FEATURES

38 SCALING DCACHE WITH RCU
As the number of processors grow, Linux either can hit a performance wall or explore new algorithms.
PAUL E. MCKENNEY, DIPANKAR SARMA AND MANEESH SONI

48 SIGNED KERNEL MODULES
Crypto techniques give device drivers a new security check.
GREG KROAH-HARTMAN

54 CONTROLLING HARDWARE WITH IOCTLS
Control all the little stuff that isn't in the UNIX programming books.
LISA CORSETTI

58 UNDERSTANDING CACHING
Use the cache right, and your code runs fast.
JAMES BOTTOMLEY

INDEPTH

69 LINUX, TALON AND ASTRONOMY
The software that controls research-grade telescopes can control your telescope too.
TONY STEIDLER-DENNISON

74 MONITORING HARD DISKS WITH SMART
Keep an eye on your drives' health with an easy-to-configure tool.
BRUCE ALLEN

78 LINUX IN AIR TRAFFIC CONTROL
Using Linux as a testing platform for mission-critical software.
TOM BRUSEHAVER

82 CONTROLLING DEVICES WITH RELAYS
Your software can break free of the box and control lights, bells and motors in the real world.
JASON ELLISON

87 TESTING APPLICATIONS WITH XNEE
Give your GUI apps a scriptable test suite with simulated X clicks and input.
HENRIK SANDKLEF

91 INTERMEDIATE EMACS HACKING
You don't have to be a LISP hacker to customize Emacs.
CHARLES CURLEY

EMBEDDED

30 PERSONAL VIDEO RECORDER BASICS
Create a custom PVR that works your way and even burns archive copies.
CHRISTIAN A. HERZOG

TOOLBOX

14 AT THE FORGE
Publishing with Bricolage
REUVEN M. LERNER

18 COOKING WITH LINUX
Scalability: from Simplicity Comes Complexity.
MARCEL GAGNÉ

24 PARANOID PENGUIN
Secure Mail with LDAP and IMAP, Part II
MICK BAUER

COLUMNS

36 LINUX FOR SUITS
Laptopia
DOC SEARLS

96 EOF
Turning IT Certification on Its Ear
EVAN LEIBOVITCH

REVIEWS

66 IBM ESERVER BLADECENTER
DANA CANFIELD

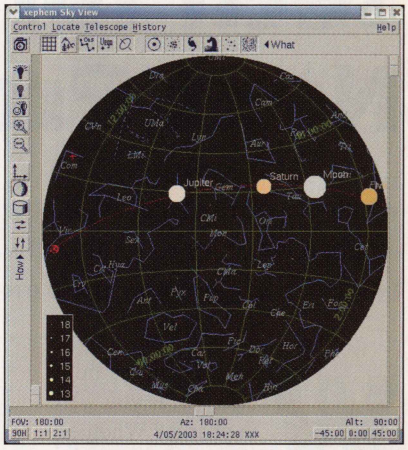
68 HACKING THE XBOX
PAUL BARRY

81 RED HAT LINUX 9 BIBLE
FRANK CONLEY

DEPARTMENTS

4 FROM THE EDITOR
6 LETTERS
10 UPFRONT
63 ON THE WEB
64 BEST OF TECHNICAL SUPPORT
65 NEW PRODUCTS
81 ADVERTISERS INDEX
95 MARKETPLACE

COVER PHOTO:
TONY STEIDLER-DENNISON



xephem provides a customized view of the night sky at your location, page 70.

NEXT MONTH

WEB-BASED BUSINESS

When you're building a machine with a half-million parts and multiple quality assurance records per part, you need work-flow management software. Giovanni Organtini and Luciano M. Barone explain the MySQL-backed system they're using to manage the construction of the Compact Muon Solenoid for CERN.

Can you run a record company without being evil? And why might a site want three different free software Web servers? Find out when John Buckman explains Magnatune, where you can get everything from Bach cello suites to electronica as MP3, Ogg or FLAC.

Tools, Tips and Tech for Your Next Project

Whether you're designing a new home theater, an observatory or security software, this issue has something for you. **BY DON MARTI**

Wouldn't things be different if you were in charge? Whether you want to archive your favorite TV shows on DVD, add an extra security check for device drivers without leaking information to marketing people or simply port Linux to your new 64-processor server, you'd do things better if you ran the place.

I've got news for you. You do run the place. You're the CEO of a multinational technology empire that has a cooperative research and development program with governments, companies and universities around the world, even your friend's hot new startup.

If you don't believe me, look at the COPYING and LICENSE files already on your hard drive. Whether you're working for yourself, starting a company or even toiling at a big company, you are free to partake in and build on the greatest information technology research effort ever. And unlike big slow "shared source" deals, you don't need to call a lawyer to plug in and start building.

Personal Video Your Way

There's no better proof of that than Christian A. Herzog's article on page 30. Want a personal video recorder that will let you make a backup? As long as the major electronics vendors design their products for cable company lawyers, you'll make the TV viewers in your family happier than the vendors ever will.

While you're watching TV, your hard drives are silently, or maybe not so silently, spinning themselves to death. With Bruce Allen's article on page 74, you can get an early warning and replace a drive on your schedule, not in the middle of the night when it fails on its own.

Kernel Power

In this special kernel issue, you'll learn that cache isn't merely a processor spec to brag about. It's a complex resource you can either use well or "blow the cache" and go as slow as main memory. Find out how Linux uses cache in James Bottomley's article on page 58.

There's plenty of other kernel innovation in this issue too. Check out Greg Kroah-Hartman's implementation of cryptographically signed kernel modules on page 48. And Paul E. McKenney is back, this time with Dipankar Sarma and Maneesh Soni, to explain a big performance win for SMP servers on page 38.

Secrets of the Software Masters

Lisa Corsetti wanted a way to check whether the Ethernet cable is plugged in. The answer opened the door to the mysteries of ioctls, and it's all explained on page 54.

Every issue, Marcel Gagné explores some new area of software for Linux, and this time he's creating simulated structures from molecules to bridges. Can you build something that will stand up? Or how many links do you need to cut to make it fall? Find out on page 18.

Don Marti is editor in chief of *Linux Journal*.

LINUX JOURNAL

JANUARY 2004
ISSUE 117

EDITOR IN CHIEF Don Marti, ljeditor@ssc.com

MANAGING EDITOR Jill Franklin, jill@ssc.com

SENIOR EDITOR Doc Searls, doc@ssc.com

SENIOR EDITOR Heather Mead, heather@ssc.com

ART DIRECTOR Garrick Antikajian, garrick@ssc.com

TECHNICAL EDITOR Michael Baxter, mab@cruzio.com

SENIOR COLUMNIST Reuven Lerner, reuven@lerner.co.il

CHEF FRANÇAIS Marcel Gagné, mggagne@salmar.com

SECURITY EDITOR Mick Bauer, mick@visi.com

CONTRIBUTING EDITORS

David A. Bandel • Greg Kroah-Hartman • Ibrahim Haddad • Robert Love • Zack Brown

PROOFREADER Geri Gale

VP OF SALES AND MARKETING Carlie Fairchild, carlie@ssc.com

MARKETING MANAGER Rebecca Cassity, rebecca@ssc.com

INTERNATIONAL MARKET ANALYST James Gray, jgray@ssc.com

REGIONAL ADVERTISING SALES

NORTHERN USA: Joseph Krack, +1 866-423-7722 (toll-free)

EASTERN USA: Martin Seto, +1 905-947-8846

SOUTHERN USA: Annie Tiemann, +1 866-965-6646 (toll-free)

ADVERTISING INQUIRIES ads@ssc.com

PUBLISHER Phil Hughes, phil@ssc.com

WEBMASTER Jeff Tinsler, webmaster@ssc.com

TECHNICAL MANAGER Dan Wilder, dan@ssc.com

***LINUX JOURNAL* IS PUBLISHED BY, AND IS A REGISTERED TRADE NAME OF, SPECIALIZED SYSTEMS CONSULTANTS, INC.**

PO Box 55549, Seattle, WA 98155-0549 USA • linux@ssc.com

EDITORIAL ADVISORY BOARD

Daniel Frye, Director, IBM Linux Technology Center

Jon "maddog" Hall, President, Linux International

Lawrence Lessig, Professor of Law, Stanford University

Ransom Love, Director of Strategic Relationships, Family and Church History Department, Church of Jesus Christ of Latter-day Saints

Sam Ockman, CEO, Penguin Computing

Bruce Perens

Bdale Garbee, Linux CTO, HP

Danese Cooper, Manager, Open Source Program Office, Sun Microsystems

SUBSCRIPTIONS

E-MAIL: subs@ssc.com • URL: www.linuxjournal.com

PHONE: +1 206-297-7514 • FAX: +1 206-297-7515

TOLL-FREE: 1-888-66-LINUX • MAIL: PO Box 55549, Seattle, WA 98155-0549 USA • Please allow 4–6 weeks for processing address changes and orders • PRINTED IN USA

USPS *LINUX JOURNAL* (ISSN 1075-3583) is published monthly by SSC, Inc., 2208 NW Market Street, Suite 407, Seattle, WA 98107. Periodicals postage paid at Seattle, Washington and at additional mailing offices. Cover price is $5 US. Subscription rate is $25/year in the United States, $32 in Canada and Mexico, $62 elsewhere. POSTMASTER: Please send address changes to *Linux Journal*, PO Box 55549, Seattle, WA 98155-0549. Subscriptions start with the next issue. Back issues, if available, may be ordered from the Linux Journal Store: store.linuxjournal.com.

LINUX is a registered trademark of Linus Torvalds.

Powerful

jet
locomotive
bodybuilder
Paracel Cyclone

Reliable

dog
firefighter
parachute
Paracel Cyclone

Flexible

clay
wire
gymnast
Paracel Cyclone

Powerful Reliable *and* Flexible

Paracel Cyclone

Paracel Cyclone

For over a decade, Paracel has been providing high-performance computing solutions for government agencies, academic and research institutions, and Fortune 500 companies worldwide. These customers rely on Paracel's proven solutions and expertise for their **mission-critical applications.**

Paracel was among the first to develop systems powered by Linux clusters to solve some of the most computationally demanding and complex problems. At Paracel, we leverage our extensive knowledge and experience while working with you to **design and build a Linux cluster** specifically for your application.

We deliver complete systems pre-configured with all of the hardware and software options and tools you'll need for seamless migration, rapid deployment, and optimal performance. Whether you need eight CPUs or a thousand, for applications ranging from computational fluid dynamics to graphics rendering or life science research, Paracel can provide a **solution that meets your unique computing needs.**

Call us today for your free expert consultation at 877.321.FAST

Paracel Cyclone

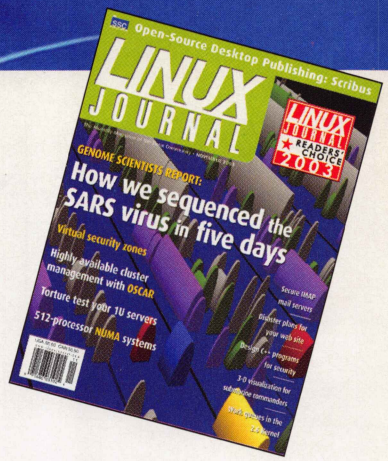
The Paracel Cyclone is a turnkey Linux cluster solution, available in flexible configurations for a range of applications. Built using only high-quality components and tested rigorously, the Paracel Cyclone provides the ultimate in performance and reliability. The Paracel Cyclone is backed by a three-year warranty and Paracel's uncompromising service and support. Contact us today for your free expert consultation at 877.321.FAST

PARACEL®

Applied High-Performance Computing

www.paracelcyclone.com

CYCLONE



Mondo and Mindi Tips

Thanks for the article about restoring with Mondo and Mindi in the October 2003 issue. It was good, but it could have included a couple more options, in my opinion. Option `-g` gives a nice graphical interface to the restoration process. Option `-l GRUB -f /dev/xxx` ensures that if one uses the GRUB loader, it will be integrated nicely. Option `-k FAILSAFE` adds compliance with Debian/GNU. I believe this last one is important for Debian users.

--
Jose Marrero

NEC Source Code?

I was very interested in your article about NEC's fault-tolerant server [*LJ*, October 2003]. I'm an IT consultant here in Buenos Aires, Argentina with some exposure to fault-tolerant and mission-critical applications. I contacted Mr John Fitzsimmons (of Aspire Communications) four or five months ago in order to help me get the source code for NEC's FT Linux. As I understand it, the software must be released under the GPL, but as of today I wasn't able to get the software. So, it was great to know from your article that "...NEC told us they are firmly committed to releasing all changes to the public under GPL." I think I will have to wait just a little longer. Do you have an approximate date for that release?

One other point, as the article stated, "At the time of this writing, NEC was reviewing and documenting its kernel changes for a planned public release, perhaps through OSDL's Carrier Grade Linux Project...", could you tell me what I should do to get the release from OSDL? Will it be available to the general public or to OSDL members only?

--
Pablo J. Rogina, IT Consultant

Dan Wilder replies: *I am not aware of any such release to date. I am copying Mr. Fitzsimmons on this note, in hope that he may have some comment.*

Electronic-Only Subscription?

Just wondering if you are going to have an electronic copy of *Linux Journal* for subscription. For readers overseas, the subscription rate is kind of expensive. I guess that's due to the shipping cost. It would be really helpful if you could provide the electronic copy as a subscription option, with the same cost as the subscription rate within the States, or cheaper.

--
Simon

You're not the first person who has asked this. We don't do this yet, but I've asked our subscription department to find out if we can.—Ed.

Mick's LDAP and IMAP Corrections

There are two errors in my article "Secure Mail with LDAP and IMAP, Part I" [*LJ*, November 2003]. First, early in the article I say (about Cyrus IMAP) "...use of databases rather than flat files to store messages has an obvious performance benefit." Actually, although Cyrus IMAP does use database files for indexing messages, it stores the messages themselves as individual flat text files.

Second, later in the article, I correctly point out that to configure Cyrus SASL on SuSE to use LDAP, you must edit the parameter SASLAUTHD_AUTHMECH in /etc/sysconfig/saslauthd. The problem is, the Cyrus SASL package in SuSE doesn't have LDAP support compiled in. You either need to build your own LDAP-enabled package with SuSE's cyrus-sasl2 source-RPM (SRPM) or forego direct LDAP support in SASL altogether and use the PAM method instead, which *is* compiled in by default. In the latter case, you need to install the module pam_ldap, and then create a file, /etc/pam.d/imap, containing these lines:

```
#%PAM-1.0
  auth        required      /lib/security/pam_ldap.so
  account     required      /lib/security/pam_ldap.so
```

Next, copy the file /usr/shage/doc/packages/pam_ldap/ldap.conf to /etc/openldap/ldap.conf, and edit it to match your environment (the most relevant settings are host, base, binddn, bindpw and TLS_REQCERT). Finally, edit /etc/sysconfig/saslauthd to include this line:

```
SASLAUTHD_AUTHMECH=pam
```

I apologize for any confusion or inconvenience that either error may have caused.

--
Mick Bauer

Busy Chef Working On Another Book

Tinyminds' Tony Brijeski (Fragadelic) recently had the pleasure of sitting down with Marcel Gagné over some coffee and asked him some questions about his recently released book *Moving to Linux: Kiss the Blue Screen of Death Goodbye* and his views on where Linux is today. He even spilled the beans on the new book he is currently working on; you'll have to read the interview to find out about it at www.tinyminds.org.

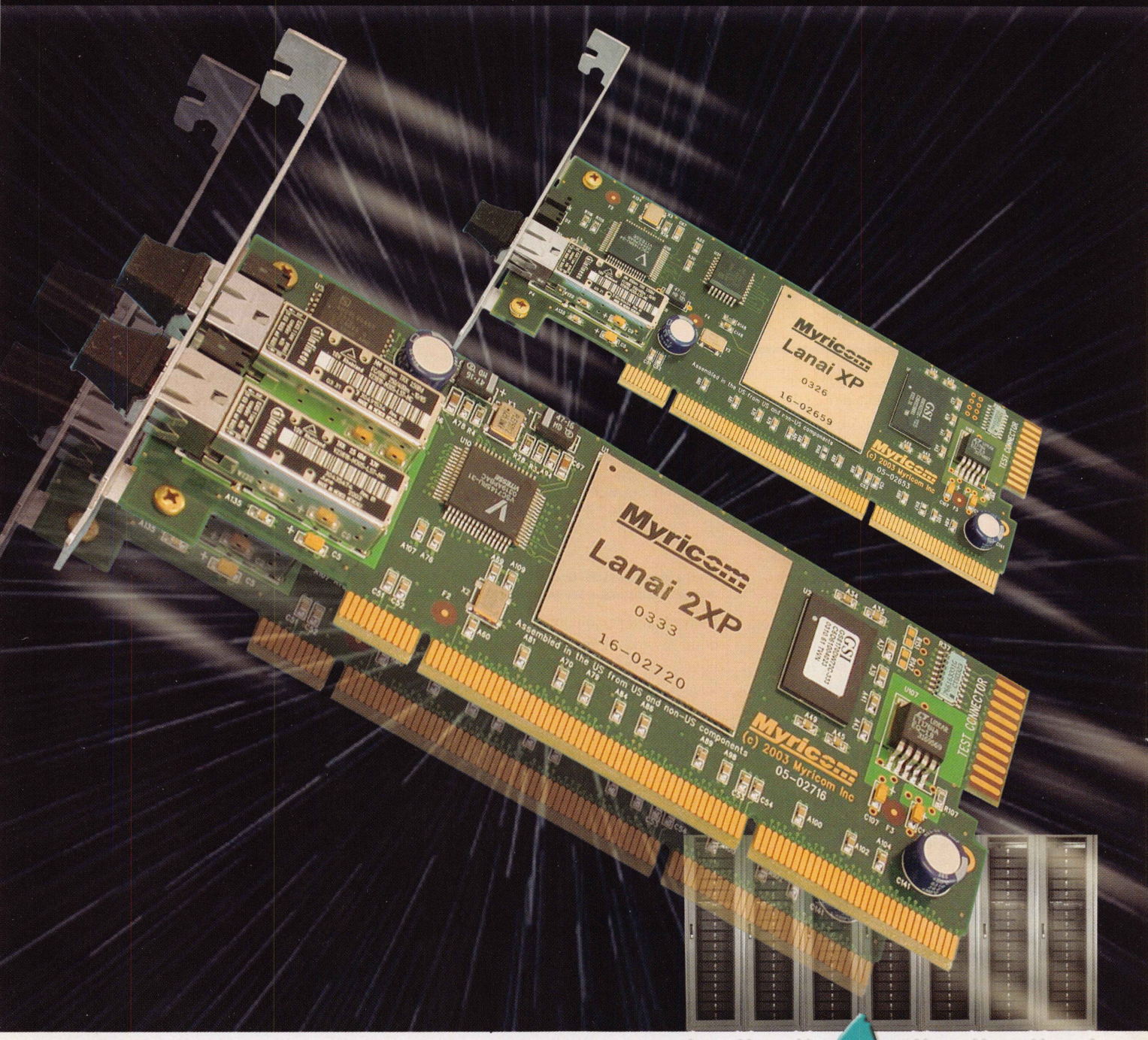
--
Mark Angeli

Modula-3 Better than C++ and Ada

This letter to the editor is a response to Dennis Ludwig's letter "Ada Isn't Awful", in the November 2003 issue. I am a counter-example to Dennis Ludwig's statement "I have yet to meet an Ada basher who really knows the language." I have programmed in Ada, mostly full-time since 1986, done serious development work on three Ada compilers and served as a resident Ada language lawyer for over ten years. It's awful, and I hate it. That said, it's not half as bad as C++, to which Ludwig primarily seems to compare Ada. I do agree with much of his cited Web article. Both languages are horribly overcomplicated to a similar degree. Most working programmers in either language don't even come close to understanding their language. At least Ada is strongly type-safe, while C++ inherited all of C's worst type-unsafe and "undefined" behaviours, so lack of understanding in Ada is more likely to show up as compile errors instead of bugs.

But the best languages, such as the Pascals, Modula-2 and the Oberons, are an order of magnitude simpler, and some of them have useful feature lists comparable to Ada. My favorite is Modula-3, with a language definition ten times smaller than Ada. It has a more flexible type-safe separate compilation, far more powerful information hiding capabilities and a choice of garbage-collected or

ASPEN SYSTEMS PROVIDES THE WORLD'S BEST BEOWULF CLUSTER SOLUTIONS



Aspen Systems offers Beowulf clusters and high-performance computing (HPC) solutions with Myricom's Myrinet-2000 PCI-X cards: the low-cost M3F-PCIXD card, and the high-end M3F2-PCIXE card. Aspen Systems' customized Beowulf clusters and HPC solutions with the new E card offer 2 fiber ports per card for double the bandwidth, so now you can have up to 1GByte/s — perfectly balanced with the 1GByte/s PCI-X bus — for a double-speed and highly reliable connection.

Myricom's D and E card PCI-X interfaces were designed to work efficiently with GM-2 and MX Myrinet software support, supplied open-source by Myricom. GM-2 is the latest evolution of a message-passing system that is used in thousands of clusters worldwide. MX is a new message-passing system that is optimized for MPI, VI, and Sockets middleware. The table shows the user-level microbenchmarks for GM-2 and MX on D and E cards in dual-Xeon hosts.

For additional information about Aspen Systems Beowulf clusters and HPC solutions with the Myrinet-2000 PCI-X cards, contact Aspen Systems' team of sales engineers by calling toll-free (U.S. and Canada only) at **800.992.9242** or **303.431.4606** or via the Internet at **www.aspsys.com**

Aspen Systems Inc.

D card Microbenchmark	***Units***	***GM***	***MX***
One-way data rate	MB/s	248	248
Bidirectional data rate	MB/s	489	489
API-level latency	µs	6	4
MPI latency	µs	6.5	4
Host-CPU overhead per message	µs	<0.5	<0.5
E card Microbenchmark	***Units***	***GM***	***MX***
One-way data rate	MB/s	>480	>480
Bidirectional data rate	MB/s	>950	>950
API-level latency	µs	5	3.5
MPI latency	µs	5.5	3.5
Host-CPU overhead per message	µs	<0.5	<0.5

Myricom AUTHORIZED RESELLER

explicitly deallocated heap objects, none of which are in Ada. Ada does non-integer fixed-point types, which is very rare. Otherwise, the feature-list differences are minor. Interested readers can read my more complete summary of the differences between Modula-3 and Ada at www.cs.wichita.edu/~rodney/languages/Modula-Ada-comparison.txt.

--

Rodney M. Bates

TALOSS Participants

In the November 2003 issue, acknowledgements were inadvertently omitted from the TALOSS (Three-Dimensional Advanced Localization Observation Submarine Software) article. Mr Ken Lima of NUWC was the originator of the TALOSS concept and the project's manager, without whose leadership this project would not be possible. The authors also would like to thank the Office of Naval Research (ONR) Code 311 (Mr Paul Quinn, Mr Gary Toth and Dr Larry Rosenblum) for the funds to complete the project.

ONR (Dr Rosenblum) funded Dr Gregory Neilsen and Mr Gary Graf of Arizona State University to develop software that computes the intersection of irregular 3-D regions within TALOSS. That software is the same software included in TALOSS as described in the subject article. A patent application for this software already has been prepared by its inventors (Graf, Nielsen, Lima and Drury).

A major portion of the TALOSS software was written at Virginia Tech University under the direction of the NUWC by graduate student, Fernando DasNeves. The funding for his work was provided under a NAVCITTI Virginia Tech grant administered by ONR (Dr Rosenblum).

The Naval Research Laboratory contributed that laboratory's expertise in 3-D rendering and visualization to the NUWC-led project. NRL (initially Mr Rob King), conducted research on 3-D interactive devices in the immersive variant of the TALOSS. Mr Douglas Maxwell, then an NRL employee, subsequently developed a 3-D grid-based approach to ocean floor rendering within TALOSS using the digital nautical chart database. Mr Maxwell also developed the collision detection algorithms used in TALOSS. Mr Maxwell's primary role, since joining NUWC a year ago, has been as a technical consultant to the project.

Mr Richard Shell of NUWC has served as the technical lead for the TALOSS Project for the past three years. Mr Todd Drury is the NUWC software development lead for TALOSS and has been responsible for the design and development of the TALOSS software for the past three years. The bottom rendering approach currently implemented in TALOSS and referenced in the *Linux Journal* article is Mr Drury's work, with Mr Maxwell's as a candidate to follow. A patent for this software was submitted several months ago.

The authors would like to express regret in the inadvertent omission of these acknowledgements. Thanks to the hard work and dedication of all involved, this project was a resounding success.

--

Douglas B. Maxwell, MSME, Research Scientist/Mechanical Engineer, Naval Undersea Warfare Center

More on Cluster Apps Please

I have had a personal nagging curiosity to build my own home cluster out of a few boxes that are readily available. Once I have it, what can I possibly do? Regarding that question, your November 2003 issue was fantastic but lacked the one thing I would like to read more about. Could you maybe consider an article describing the different applications that are available for clusters?

--

S. B.

Linux Training?

As a self-taught Linux user, I really enjoy and appreciate the information that gets published in *LJ*. I am wanting to learn more and am looking for courses in Linux, but only seem to find the occasional Linux conference or workshops that offer Linux certifications. Being Linux-certified is my goal, but I know I still require classroom study. Can you point me in the right direction? Thanks! Keep up the good work!

--

Jason Fales, Dallas, Texas

Try the directory at lintraining.com.*—Ed.*

fgets Good, gets Bad

I enjoyed Cal Erickson's article, "Writing Secure Programs", in the November 2003 issue, and definitely plan on looking at FlawFinder and other tools. One nit: "Do not use fgets when reading data, as this allows overflows." I think this was a typo; it should have read "gets...allows overflows", because fgets has a length parameter.

--

Collin Park

You're right. As soon as you finish that "Intro to C" class that requires you to use gets, you just don't use it anymore.—Ed.

License for Bioinformatics Software

In the November 2003 cover article, "Sequencing the SARS Virus", Krzywinski and Butterfield refer to phred, phrap and consed as the "open-source workhorses" of the Human Genome Project. Although these programs are indeed bioinformatics workhorses, and although the no-fee academic licensing terms under which they are available are generous by proprietary software standards, the terms under which they are distributed do not conform to the Open Source definition at least to the extent that they discriminate against fields of endeavor. The terms apply to academic or nonprofit use but not to business use (Open Source definition version 1.9, section 6).

BLAST, another workhorse mentioned in the article, has diverged into two main branches, one in the public domain (which can be obtained from the National Center for Biotechnology Information [NCBI]) and one distributed in a fashion similar to phred, phrap and consed.

--

D. Joe Anderson

Errata

October 2003, page 88: *in Listing 1 of the article "Building a Linux IPv6 DNS Server" by David Gordon and Ibrahim Haddad, there should have been a closing brace after the first two lines.*

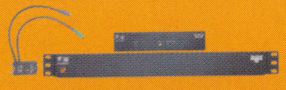
November 2003, page 50*: in the Command-Line Bionformatics Sidebar of "Sequencing the SARS Virus" by Martin Krzywinski and Yaron Butterfield, the* `t` *in the third line of code should be* `tr`.

We welcome your letters. Please submit "Letters to the Editor" to ljeditor@ssc.com or SSC/Editorial, PO Box 55549, Seattle, WA 98155-0549 USA.

Is security this big when managing your data center?

Ours is, only in smaller packages.

Fingerprint Authentication Scanner
AlterPath™ Bio



Enterprise KVM Solutions
AlterPath™ KVM

Advanced Console Server
AlterPath™ ACS

Network Management Gateway
AlterPath™ Manager

Intelligent Power Distribution Unit
AlterPath™ PM

Cyclades' data center management solutions offers a full range of security features across its entire product line of console servers, power management, KVM, biometric scanner and network management. With SSH v2, IP Filtering, strong authentication, event logging and data logging, Cyclades can make your network into a secure heavyweight contender in the data center world.

LINUX INSIDE

For a FREE white paper on data center security, please visit us at www.cyclades.com/securitywp

LINUX JOURNAL READERS' CHOICE 2003
"Best Hardware for Linux since 1995"

www.cyclades.com/lja
1.888.cyclades ▪ sales@cyclades.com

cyclades
Everywhere with Linux

©2004 Cyclades Corporation. All rights reserved. All other trademarks and product images are property of their respective owners. Product information subject to change without notice.

/usr/src/linux/CREDITS

Linux kernel hackers at the 2003 Kernel Developers' Summit in Ontario, Canada, photographed by Richard Guy Briggs and submitted by Robert Love.

milter-sender:
www.snert.com/Software/milter-sender

milter-sender is a Sendmail milter plugin that connects back to the sender of a message as it's being received. Basically, if the sender doesn't exist on the sender's alleged primary mail server, the message is almost certainly spam and is rejected. This plugin has reduced my spam significantly. And, although the arms race isn't over, I always can add more milter plugins. I'm now running milter-sender and SpamAssassin via milter. Requires: Berkeley DB, Sendmail with milter, libsnert and glibc.

—DAVID A. BANDEL

LJ Index—January 2004

1. Number of changes to the Linux 2.6 kernel in development from 2002 through mid-September 2003: 12,479
2. Number of merges in the kernel, not counting apply-by-proxy patches: 2,386
3. Approximate average number of daily merges: 9
4. Number of individual developers working on the kernel: 554
5. Percentage of all changes done by the top 10% of kernel developers: 78
6. Percentage of all changes done by the top 20% of kernel developers: 90
7. Linux percentage of the paid server operating environment (SOE) category in 2002: 23.1
8. Microsoft percentage of the paid SOE category in 2002: 55.1
9. Millions of new licenses shipped in the SOE environment in 2002: 5.7
10. Linux percentage share of the client operating environment (COE) category in 2002: 2.8
11. Macintosh percentage share of the COE category in 2002: 2.9
12. Percentage share of the COE category in 2002 for other OSes: 0.5
13. Microsoft percentage share of the COE category in 2002: 93.8
14. Number of operating systems other than Linux and Windows that grew in revenue in 2002: 0

Sources

1–6: Linus Torvalds

7–14: International Data Corp.

They Said It

There are a lot of reasons for the penguin. I was bitten by a penguin. And it's a true story. It's funny, because there are a lot of Web sites about the penguin. There's The History of Tux (www.sjbaker.org/tux) and things like that. And some of these Web sites have some of my explanation. And they almost universally say, "It's a great story, but it's not true."—that I was bitten by a penguin.

It's true! I was bitten by a penguin! I mean, really! Take it from me! I'm wounded!

Okay, so he wasn't six foot tall.

—LINUS TORVALDS, FROM HIS TALK ON THE LINUX LUNACY GEEK CRUISE, SEPTEMBER 2003

We're very aggressive around migrating to Linux; the target is to buy no new UNIX kit....Until one year ago most mainstream applications were on UNIX. The back-office systems are on mainframes, but everything else was on Sun.

—LEHMAN BROTHERS CIO EMEA RUSSELL (www.cbronline.com/magazine/2d7f2e75fecd8bbfc2256d7800317032)

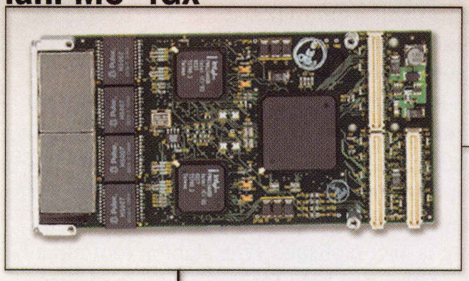
Anything touching Linux.

—GAP CORP. CIO KEN HARRIS, ON HIS COMPANY'S TECHNOLOGY PROJECT PRIORITIES (asia.cnet.com/newstech/systems/0,39001153,39154483,00.htm)

Linux *On Demand*

You Decide...Linux ready on every board

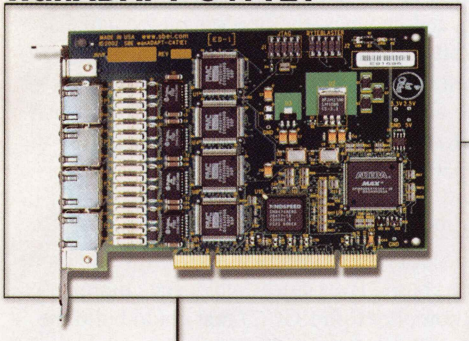
lanPMC-4Gx



Quad port Gigabit Ethernet copper or fiber PMC adapter

- 2 or 4 auto-negotiating 10/100/1000 Ethernet ports
- Fiber version supports long and short reach
- Auto sensing speed, simplex/duplex, flow control, and detection of polarity and cable lengths on copper version

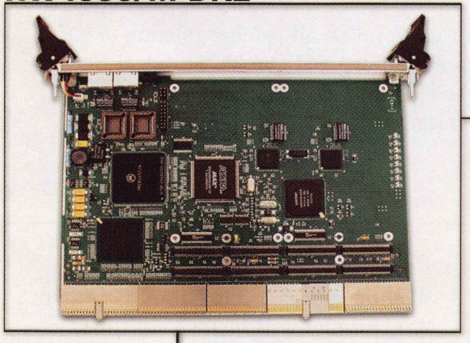
wanADAPT-C4T1E1



Channelized quad port T1/E1/J1 PCI WAN adapter

- 128 channels (DS0)
- Line buildouts from DSX to DS1
- Supports D4, SF, ESF, SLC-96, JT-6704 framing
- Selectable clock options

HW400c/M DKL



6U CPCI processing blade with dual PTMC sites

- 333 MHz PowerPC CPU
- PICMG 2.16 Packet Switching Backplane enabled
- Dual 10/100 Ethernet ports
- Includes TimeSys Linux GPL Software Developer's Kit
- Drivers for T1/E1, T3/E3, HSSI, V.35, X.21 & other SBE PMCs

SBE®

WAN • LAN • Storage • Carriers • Custom

flexibility on demand | 925-355-2000 | info@sbei.com | www.sbei.com

diff -u: What's New in Kernel Development

Intel has come out with some event notification patches that provide profiling hooks at process termination, in addition to the already existing hooks provided at process creation. Intel's work should be useful for enhancing such profiling tools as Oprofile, Perfmon and VTune. Although adding hooks like these potentially can provide a way for proprietary software vendors to bypass the GPL licensing restrictions, it does not appear that Intel's work suffers from that problem. **Frederic Rossi** also has been working on his own unrelated event notification code, called **asynchronous event mechanism** (AEM). AEM allows general-purpose user-space applications to register for system events by providing their own sets of event handlers [*LJ*, July 2003]. The main goal for this is to be able to pass data quickly from the kernel back to user space. Although Intel and AEM's work are not directly related, they reveal an interesting trend in Linux development toward greater and greater communication between the layers of the system.

The DSI Development team has produced a standalone module, called **digsig**, that checks the embedded digital signature of an executable program before running it. If the file has been modified, digsig prevents its invocation. A wide variety of applications could make use of this—for everything from worm protection to digital rights management. Unlike Trusted Computing's attestation feature, digsig's verification process remains under the control of the machine owner. It is not possible, for example, for remote systems to rely on the digsig module on your system to verify that your computer is running particular versions of particular software that those remote systems have approved.

Apparently, all **AMD Athlon** processors suffer from a bug that can cause prefetch instructions to receive memory management faults for certain addresses under certain conditions. This is similar to the infamous f00f Pentium bug, in that both are bugs within the hardware itself that the OS simply must work around. The good news is that, in most cases, the bug doesn't do any actual harm. Triggering the bug is rare, which may account for why it's gone undetected for so long. **Andi Kleen** has been working on a Linux workaround for 2.4 and 2.6, and the AMD folks have affirmed that his workaround will be forward-compatible with future Athlons that have the bug corrected in hardware. However, according to Andi, some Athlons, including the K7, may never be fixed, and so his workaround may be needed on new systems for quite a long time.

SGI has released a GPLed driver for its **Altix** serial console, but in the copyright notice, SGI explicitly reserves the rights to any patents used in that code. Although the Free Software Foundation has confirmed to SGI that this is not technically a violation of the GPL, it could result in a situation in which code from a GPLed project (like the kernel) would not be allowed to be used in other GPLed projects. The question then becomes, should the code be allowed into the kernel under those conditions? **Alan Cox** has decided to investigate the situation with both SGI and the FSF to confirm that the FSF really doesn't consider it a violation of the GPL and to see if SGI is willing to modify its position.

Chris Rivera and **Robert Love** have released the **slabtop** standalone utility, a tool to monitor the allocation and freeing of memory in the cache. The kernel maintains a cache of allocated memory that it disburses to processes as needed. But sometimes this memory needs to be freed, and the slab is the way that such freeing is accomplished. The slabtop tool monitors each slab cache, presenting data for that slab in real time, and it also concurrently updates a set of global statistics on all slab activity on the system.

Matthew Wilcox has begun separating kernel headers into two categories, those that should be visible to user-space sources and those that shouldn't. It's always a problem when a user application comes to rely on the wrong kernel headers. As the kernel grows and changes, the headers may change as well, and the user software may break. To recover from this, a strict separation must be maintained. Matthew's work helps make this easier. He also managed to inspire other folks, such as **Erik Andersen**, to start cleaning up some type naming conventions, in preparation for 2.6.

A new division of kernel code has been initiated, similar to the arch/ directory. But instead of dividing code on an architecture-by-architecture basis, the new division works on a compiler-by-compiler basis. Ever since Intel's **ecc** compiler became the second compiler (after GCC) that could build the Linux kernel successfully, subtle conflicts have crept in, causing developers to start submitting patches specifically to handle particularities of one compiler or the other. After some recent debate over support for in-line assembly, which GCC supports but ecc does not, Linus Torvalds decided to create per-compiler directories to house all patches relating to particular compilers. **Sam Ravnborg** and **David Mosberger** apparently have decided to lead that effort, and they already have submitted patches to initiate the transition.

—ZACK BROWN

They Said It

It is very very hard to find people who don't flame and are calm and rational—*and* have good taste. I mean it's like...give me one honest man. It doesn't happen...too much. And at the same time, when it happens, it matters a lot. Just a *few* of these people make a huge difference.

—LINUS TORVALDS, ALSO FROM HIS TALK ON THE LINUX LUNACY GEEK CRUISE

In the early days, Linux was under the wire without the chief information officer's (CIO) knowledge. It was the technical decision for things such as a file and print server, with cost or reliability as the driver. Now we have CIO adoption.

—ADAM JOLLANS, IBM WORLDWIDE LINUX SOFTWARE MARKETING STRATEGY MANAGER (www.vnunet.com/News/1144291)

A slide rule is a precision instrument. A good one will last a long time, in fact, for a lifetime unless you abuse it. You've got to look ahead for 10, 20, 30 or even 40 years if you want to be sure you are getting your money's worth.

—"HOW TO CHOOSE A SLIDE RULE" BY DON HEROLD (1940 KEUFFEL & ESSER CO. BROCHURE POSTED ON mccoys-kecatalogs.com)

Rackspace — Managed Hosting backed by Fanatical Support.™

Servers, data centers and bandwidth are not the key to hosting enterprise class Web sites and Web applications. At Rackspace, we believe hosting is a service, not just technology.

Fanatical Support is our philosophy, our credo. It reflects our desire to bring responsiveness and value to everything we do for our customers. You will experience Fanatical Support from the moment we answer the phone and you begin to interact with our employees.

Fanatical Support has made Rackspace the fastest-growing hosting company in the world. Call today to experience the difference with Fanatical Support at Rackspace.

LINUX JOURNAL READERS' CHOICE 2003

Thanks for honoring us with the 2003 Linux Journal Readers' Choice Award for "Favorite Web-Hosting Service"

rackspace™ MANAGED HOSTING

1.888.571.8976 or visit us at **www.rackspace.com**

Publishing with Bricolage

Streamline your Web site with a professional newspaper-like publishing model. With Bricolage, you can set up approvals and keep track of who's editing what. **BY REUVEN M. LERNER**

If you publish an on-line newspaper or magazine, you probably already have looked into a content management system (CMS). CMS software makes it easy to keep track of the many pages on a Web site by handling issues as varied as users, groups, permissions, editing responsibilities and site-wide templates. The more pages on your site, or the more people working on it, the more likely it is that you can benefit from a CMS.

Many companies have sprung up to meet this need, offering CMS software that claims to do everything but write your site's content. (Although given the quality of writing on some sites, you sometimes have to wonder.) It has taken some time, but a number of open-source CMS packages are available that can fulfill even the most complex site requirements. One of the most powerful and popular packages is Bricolage, which combines the mod_perl module for Apache, the PostgreSQL relational database and the HTML::Mason templating system into one neat package (sourceforge.net/projects/bricolage).

In previous installments of this column, we looked at some basic configurations of a Mason system, including alerts that can be set to fire when certain events occur. Until now, however, we have neglected the most important task of all for a CMS, namely, publishing content to an actual Web site. This month, we follow an article as it goes from inception to publication, working its way down the Bricolage publishing pathway.

Creating a Site

The first step in all of this is to create a Web site on which the content can be published. I created a virtual host named output.lerner.co.il on my server, with its own directories for error and access logs. I then added an appropriate VirtualHost directive in my Apache configuration file, as follows:

```
<VirtualHost 69.55.225.93>
    ServerName output.lerner.co.il
    ServerAdmin reuven@lerner.co.il
    DocumentRoot /usr/local/apache/
    ➥v-sites/output.lerner.co.il/www
    CustomLog /usr/local/apache/v-sites/
    ➥output.lerner.co.il/logs/access-log combined
    CustomLog /usr/local/apache/v-sites/
    ➥output.lerner.co.il/logs/referer-log referer
    ErrorLog /usr/local/apache/v-sites/
    ➥output.lerner.co.il/logs/error-log
</VirtualHost>
```

Now, when my server receives a request for output.lerner.co.il, it looks in the output.lerner.co.il/www directory rather than in the main document root, generally defined to be /usr/local/apache/htdocs.

Before we can publish any articles to the Web with Bricolage, we must tell the CMS where new files should be deposited. In Bricolage, this is done with the Destinations menu option, under the Distributions heading. You can create multiple output destinations, allowing you to have multiple sites with a single Bricolage instance. This might be the case for a publisher whose staff produces several different newspapers. Each output destination can be on the local filesystem or on a remote site accessed by FTP.

Click on New Destination to create a new publishing destination. Most small sites need only a single channel, which allows all documents to be exported to the site's DocumentRoot on the disk. You then must indicate a few things: 1) whether the Web site resides on the same computer as the Bricolage system or whether files must be copied to a remote computer using FTP and 2) the output channels to which this should be made private. We assume the Bricolage server and the final Web server are on the same computer—but in a larger-scale environment, especially where performance is an issue, it would be wise to split them up. The first new destination screen is shown in Figure 1.

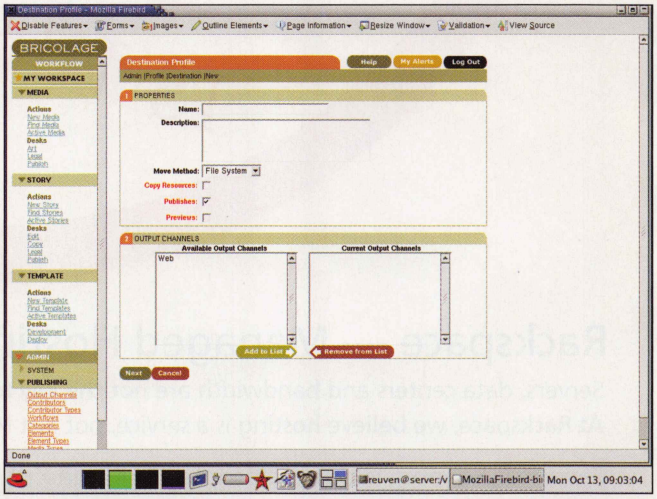
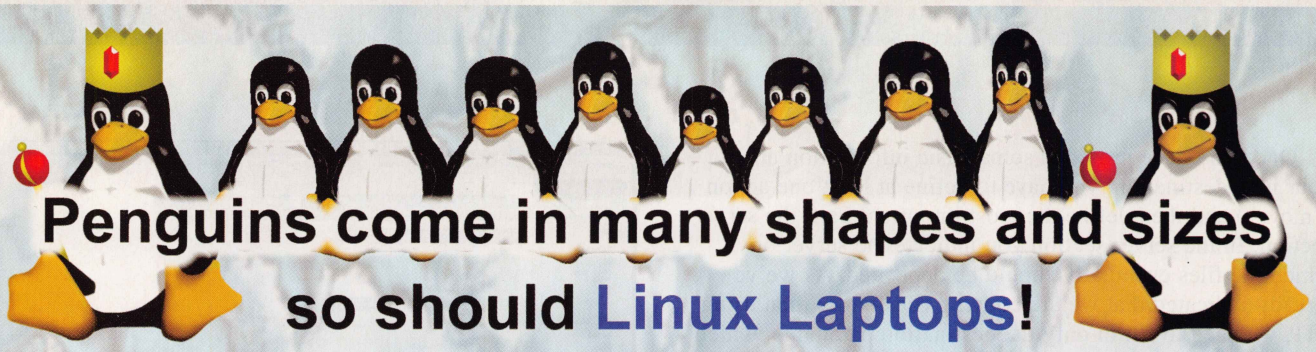


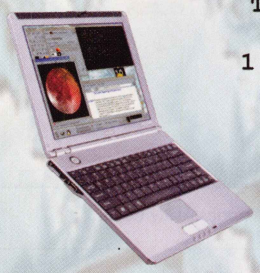
Figure 1. Setting Up a New Destination

Although Bricolage allows you to define new output channels, keep the default Web channel for now. The difference between output channels and destinations can be a bit confusing at first. Think of an output channel as a logical target and a destination as a physical target, and then realize that you can mix and match them in all sorts of different ways. You can have multiple output channels going to a single destination or a single output channel going to multiple destinations.



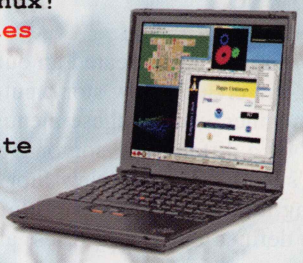
Penguins come in many shapes and sizes so should Linux Laptops!

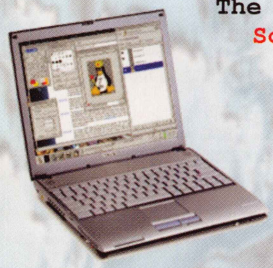
You are invited try out our laptops in Booth 138 at the Linux World Expo in New York from January 21-23, 2004.



The Meteor: 2 lb Linux!
Sharp Actius MM10
1.0 GHz Transmeta Crusoe
15 GB HD
256 MB RAM (Fixed)
802.11b Wireless
10/100 Ethernet
USB Cradle
0.5" Thin!

The Raven: 3.5 lb Linux!
IBM ThinkPad X Series
1.3-1.4 GHz P-M
20-40 GB HD
256-2048 MB RAM
APM Suspend/Hibernate
802.11b Wireless
10/100 Ethernet
FireWire
1" Thin!





The BlackPerl: 4 lb Linux!
Sony Vaio V505 Series
2.0-2.4 GHz P4-M/
1.4-1.8 GHz P-M
40-80 GB HD
256-2048 MB RAM
CDRW/DVD
FireWire
802.11b Wireless
10/100 Ethernet

The Toucan: 5 lb Linux!
IBM ThinkPad T Series
X@1400x1050
1.3-1.7 GHz P-M
20-80 GB HD
256-2048 MB RAM
CDRW/DVD
10/100 Ethernet
802.11b Wireless
APM Suspend/Hibernate

The Gazelle: 8 lb Linux!
Sony VAIO GRT Series
X@1600x1200
2.0-2.8 GHz P4-M
40-80 GB HD
256-2048 MB RAM
CDRW/DVD or DVD+/-RW
10/100 Ethernet
Memory Stick/Jogdial
FireWire

The Rhino: 7 lb Linux!
Dell Inspiron 8500/
Latitude D800
X@1920x1200
2.0-2.6 GHz P4-M/
1.6-1.7 GHz P-M
30-80 GB HD
256-2048 MB RAM
CDRW/DVD, DVD, CDRW,
or DVD+/-RW

Since 1999, EmperorLinux has provided pre-installed Linux laptop solutions to universities, corporations, and individual Linux enthusiasts. We specialize in the installation and configuration of the Linux operating system on a wide range of the finest laptop and notebook computers made by IBM, Sony, and Dell. We offer a range of the latest Linux distributions, as well as Windows dual boot options. We customize each Linux distribution to the particular machine it will run upon and provide support for: ethernet, modem, wireless, PCMCIA, USB, FireWire, X-Server, CD/DVD/CDRW, sound, APM, and more. All our systems come with one year of Linux technical support by both phone and e-mail, and full manufacturers' warranties apply. Visit www.EmperorLinux.com or call 1-888-651-6686 for details.

Successful Linux laptop solutions found for:

Sandia Nat Lab DOE NASA JHU Applied Physics Lab Max Planck Inst Woods Hole NRAO Naval Research Lab Smithsonian Institute Los Alamos Nat Lab UC Lawrence Berkeley Labs NOAA Lawrence Livermore Nat Lab US Army Research Lab U Penn U Utah U Maryland Yale Cornell Columbia U Oregon NC State Duke U Massachusetts George Mason Colorado State Caltech U Illinois Georgia Tech U Minnesota U Chicago Pace U UCSF UCSD Rutgers Princeton NYU Purdue Boston U MIT U Arizona George Washington U U Hawaii U Maine

www.EmperorLinux.com 1-888-651-6686

Model prices, specifications, and availability may vary. All trademarks are the property of their respective holders.

Once you have provided some basic information about your new destination, you have to define at least one action ("move") and then define the most important part, the server section. In the server section you indicate the server on which the files eventually should be deposited. As we've already indicated that we are copying files rather than using FTP, we need to fill in only the destination pathname, which should match the DocumentRoot of the Apache virtual host where the document will be available.

Writing and Publishing

Now that we have created an output channel, we can go back and create a new story, putting on our reporter hat. (Remember, each of these menus normally is meant to be used by a different person or group of people on a magazine's staff.) Click on New Story in the the Story menu, giving it the minimum of a title, a category (/, unless you have defined other categories) and a priority (normal). You also can set the publication date, which indicates not only when the story should be published to the Web, but also the URL with which it should be available. Longtime readers of Salon.com probably have noticed that each story's URL contains the date of publication; it should come as no surprise that Bricolage, which evolved from the CMS developed at Salon, continues in that tradition.

When you have filled in this basic information, click on the Create button. You now have the opportunity to create the text of your article, adding paragraphs (one at a time or separated by blank lines using the bulk Edit button). When your story is finished, you can submit it to the Edit Desk using the Check In button at the bottom of the page, sending it to the Edit Desk.

Unfortunately, the nature of a Web-based interface and the flexibility that Bricolage offers means that the interface for Bricolage often is a bit complex, forcing you to work with a number of different Submit buttons. Fortunately, the buttons are labeled clearly, which means that even if you end up saving your story rather than checking it in to the Edit Desk, you probably aren't going to delete the story accidentally. That said, new Bricolage users would be wise to read the labels on the buttons before assuming that all Submit buttons are created equal.

Now that we have submitted the article to the Edit Desk, we change hats once again and pretend to be the editor who has to review the story. Click on the Edit menu under the Desks subheading in the Stories menu, and you get a list of all stories currently checked in to the edit desk (Figure 2). Stories already published have a P in their titlebars. Stories checked out (that is, available for changes by an editor) can be edited. But we're not going to edit the story right now, because we know that the author is one who consistently finishes his sentences and carefully double-checks his work. For now, we send it along to the Publish Desk, where things are reviewed one last time and then sent to the Web. We do this by choosing Publish on the Move To menu and then clicking Move Assets.

We then can go to the Publish Desk, where items are reviewed one final time before they are published to the Web. Notice that Bricolage comes with built-in support for copyediting and legal reviews. You can create new desks as well, if your organizational structure is more complicated than the default setup.

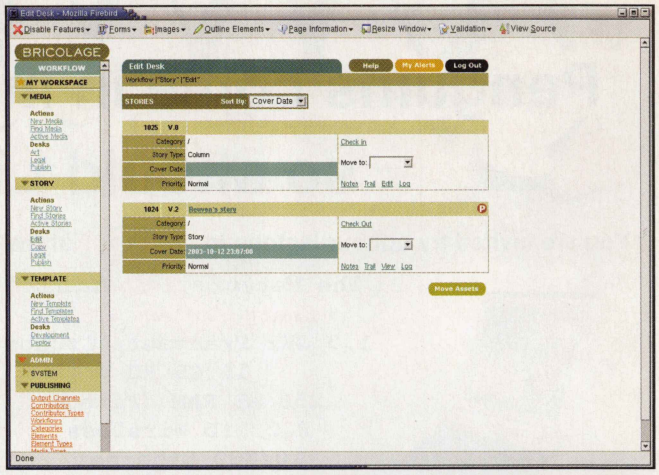


Figure 2. At the Edit Desk, you can check for stories ready to edit.

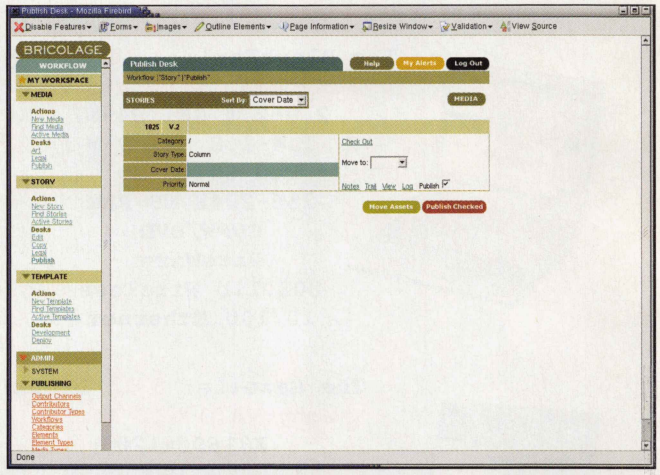


Figure 3. Review stories for publication at the Publish Desk.

In many ways, the Publish Desk is the same as the others; it allows you to add notes to a story, view a story's edit trail or even see the log of changes. You also can edit the story at the Publish Desk by checking it out and clicking on the Edit link. The most important parts of the Publish Desk, however, are the additional check box (marked Publish, not surprisingly) and a button marked Publish Checked (Figure 3).

When you click on Publish Checked, Bricolage gives you the option of indicating when the story should be published. This is different from the publication date of the story. You might want a story with a February 1st date, for example, to be available on January 31st. Alternatively, you might want it to be available only on February 7th. Clicking on the Publish Assets button puts this process in place, with Bricolage reporting it has published the story (or stories) in question. And sure enough, our document has been published to the URL we assigned it earlier—in this case, as index.html in the DocumentRoot.

Management Issues

When a print newspaper or magazine publishes a story, there is no way to change what has been displayed. But on the Web, all you need to do is modify the story in question, and everyone immediately sees the changes. Ignoring the issues of journalistic ethics raised by this technological ability, Bricolage makes it easy to correct a story and then redeploy it on the site.

To modify an already-published story, go to the Active Stories menu item, which brings up a list of stories currently available. You can choose to check out a story and make it available for editing by the current user. (Bricolage's version-control system ensures that only one person can edit a file at a time.) You then can edit the file and send it to the Edit Desk, just as you did when you originally wrote the story. And as before, the Edit Desk must pass it along to the Publish Desk; from there it can be published to the Web.

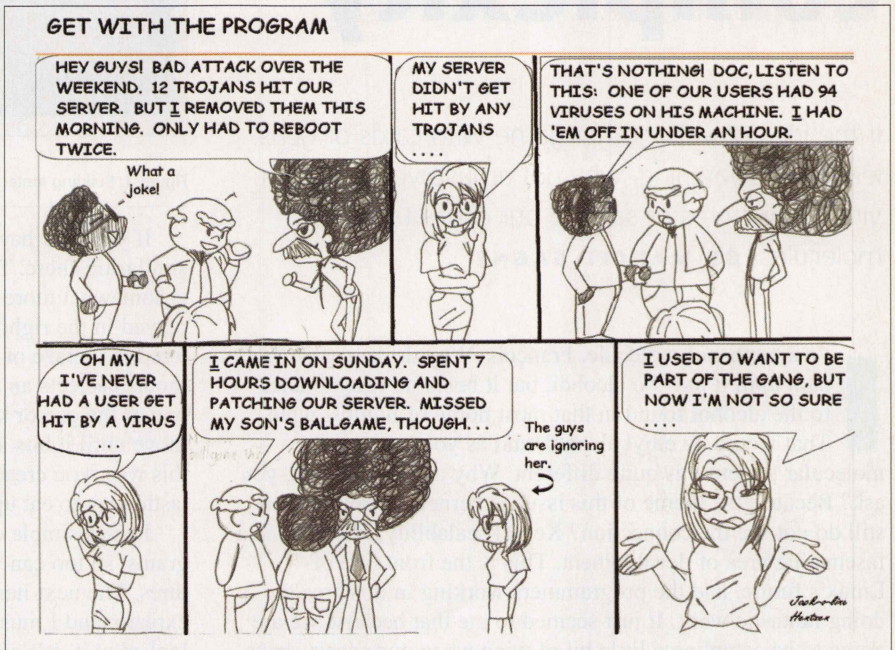
If you have worked on only small Web sites before, then this task might appear to be far too complicated. After all, isn't part of the Web's beauty the fact that we can change documents and immediately see the results of those changes? It's true that a content management system introduces a set of trade-offs into the workings of a Web site. No longer can you simply modify a file on the disk; you must log in, check the file out, make the modifications, check it back in and send it to the Publish Desk, which might even reject your changes.

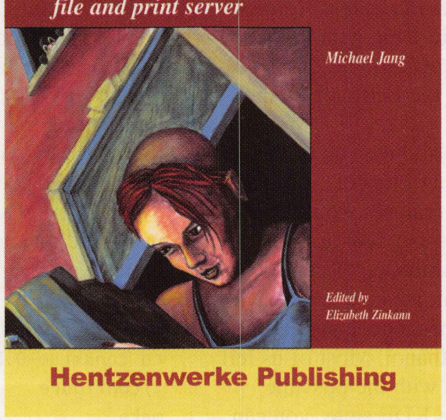
But complicated and bureaucratic as this procedure might be, it is better than the alternative, which is multiple people fighting over ownership of a file or different people trying to edit a file—sometimes in different ways—on a live site. A CMS is designed to slow you down, much as a seat belt is designed to prevent you from moving too much. But in both cases, the restrictions are designed to help you by stopping you from hurting yourself. Moreover, this bureaucracy can save you in a pinch, because every previous version of the document always is available. So if you accidentally removed half of an important story, you can revert to the previous version that Bricolage has stored in its database.

Conclusion

Bricolage is a complete, sophisticated open-source content management system with an impressive array of features. As we saw this month, we easily can set up a Bricolage-based site and publish stories through a distribution channel that we created with a simple set of virtual host definitions in our Apache configuration. Next column, we conclude our discussion of Bricolage by looking at ways in which we can customize the output templates, turning a simple black-and-white publication into something with more pizzazz.

Reuven M. Lerner, a long-time Web/database consultant and developer, now is a first-year graduate student in Learning Sciences at Northwestern University in Chicago, Illinois. You can reach him at reuven@lerner.co.il.





ISBN 1-930919-46-8 298 pages $49.95

Need to move your file server to Linux in a hurry?

"*Linux Transfer for Windows Network Admins*" will show you how to convert your existing Windows PDC and file/print server to Linux, using your knowledge and skills as a Windows admin to get the job done quickly and easily.

Download free sample chapters and order online at www.hentzenwerke.com

Also available at selected bookstores nationwide

Hentzenwerke

Scalability: from Simplicity Comes Complexity

If the idea of yet another game with cards or guns leaves you yawning, why not amuse yourself with virtual structures or sample our chef's tasty molecules? **BY MARCEL GAGNÉ**

It's a methanol molecule, François. Why do you ask? *Non, mon ami*, it may be alcohol, but it bears no resemblance to the alcohol found in that most noble of liquids, wine. That would be ethyl alcohol, and as you can see, its molecular structure is quite different. Why am I doing this, you ask? Because the theme of this issue is kernel scalability. You still do not see the connection? Kernel scalability, François, is a fascinating area of development. This is the front line of Linux's future, and the programmers working in this area are doing fantastic work. It just seemed to me that because we are going to be sampling a little bit of wine when our guests arrive, a different kind of scalability might be in order. François, why aren't you paying attention?

Ah, our guests are here! Welcome, *mes amis*, to *Chez Marcel*, home of fine Linux fare, exceptional wines and from-the-ground-up design. François and I were discussing the theme of this issue. Please sit; your tables are ready and my faithful waiter will run to the cellar *immédiatement* to fetch the wine. I think the 2001 Châteauneuf-du-Pape with its rich fruit, tantalizing texture and complex aromas should do nicely. You'll find it in the south wing next to the entrance to the reading room.

While we wait for the wine, let me introduce you to the first item on today's menu. As I was telling François, I felt that because we are sampling a little *vin*, our features would explore scalability of a different sort. Just as a few lines of code can have a huge impact on the performance of certain kernel functions, so too can a few stray atoms have a massive effect on a final molecular product. For instance, consider the difference between ethyl alcohol and methanol. Most of us left chemistry behind a number of years ago, with the possible exception of cooking, *non*? As part of the KDE suite, you can find a little program from Andreas Wüst called *KAtomic* (Figure 1). It is listed under the Games menu along with other strategy or puzzle games.

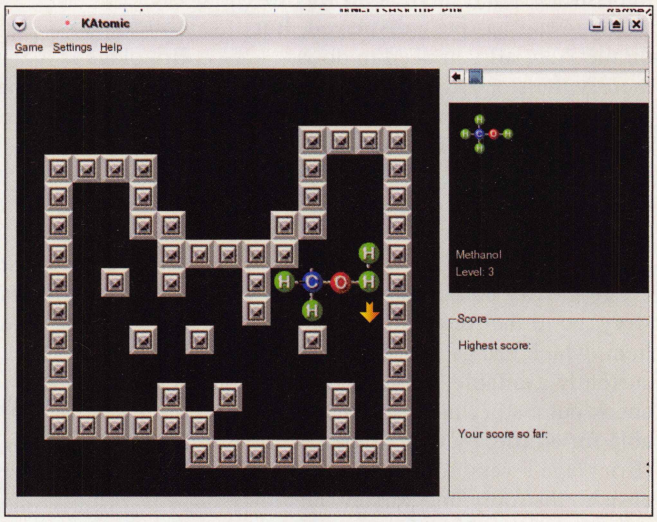


Figure 1. Building Molecules Atom by Atom with *KAtomic*

If you ever have played *Sokoban*, you should notice some similarities here. The premise of *KAtomic* is simple. Execution is somewhat more complex. You are given a molecule, displayed in the right-hand pane, with its component atoms scattered in a maze of sorts. Pressing the Tab key (or clicking the mouse) selects an atom that you then set in motion by pressing one of the cursor keys. Once in motion, the atom continues to move until it hits a barrier, either a wall or another atom. In this way, you create complexity from simplicity. It's also a fantastic way to eat up your free time.

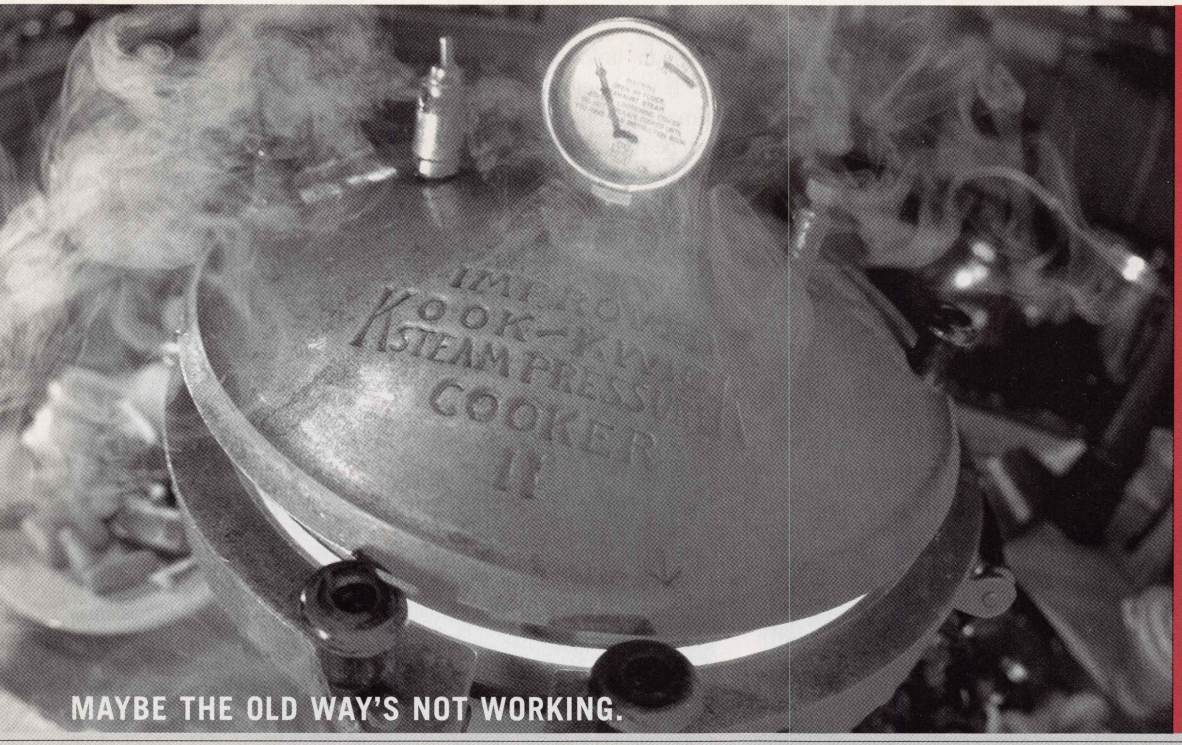
Just as simple code is used to build ever more complex programs, so too can we move from molecules to visible structures. The next item on tonight's menu is a little hard to explain, and I must admit that I had some doubts when I first looked at it. It's called *Construo*, and quite frankly, it turned out to be an amazingly addictive little program.

Ingo Ruhnke's *Construo* is, in effect, a two-dimensional construction program that lets you build objects on-screen using rods and springs connected at points that you define. The resulting structure then can be run, which essentially means letting real-world physical forces act on it. The idea is to build a stable structure, which is a lot harder than it sounds. If your creation starts to waver and finally crashes to the ground, it's time to rethink and modify your construction. A note on the *Construo* Web site makes a point of letting us know it is "currently not a real game", but I beg to differ. To get in on this thought-provoking bit of fun, head over to the *Construo* Web site at www.nongnu.org/construo.

Building *Construo* simply is a matter of extracting, configuring and making. Yes, *mes amis*, it's the classic extract and build five-step:

```
tar -xzvf construo-0.2.2.tar.gz
cd construo-0.2.2
./configure
make
su -c "make install"
```

DATA CENTER ABOUT READY TO BLOW?



Rackable Systems servers—now with Distributed DC Power—improve thermal management, reliability and power efficiency.

GREATER RELIABILITY & MINIMIZED MAINTENANCE

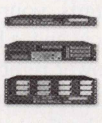
Now fueled by Distributed DC Power, Rackable Systems server clusters are more reliable, flexible and efficient than the competition. By distributing redundant DC Power to each server—and replacing the standard AC power supply with a 70 times more reliable DC card—availability is significantly improved and servers are easier to maintain. They also run cooler, harnessing the full power of the fastest Intel® Xeon™ processors. Best of all, Rackable Systems DC clusters accept either AC or DC input power for easy, immediate deployment. All at a price comparable to standard AC-based solutions.

THERE IS A BETTER WAY

As pioneers of large-scale data center solutions, Rackable Systems continues to lead the industry with innovative server designs for Biogen, Electronic Arts, Google, Inktomi, Lawrence Livermore National Laboratory, nVidia, SLAC, Tellme Networks, Webex and Yahoo!. Our open architecture configurations are built-to-order to ensure maximum flexibility and ease of serviceability. Our innovative cabinet design features twice the density at a fraction of the cost, while optimizing thermal efficiency. And with Rackable Systems' new Distributed DC Power technology, you've got a solution that can't be beat.

Rackable® systems

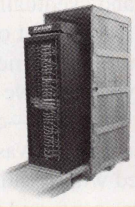
Flexible Configurations



- Customized & built-to-order
- Open architecture
- Superior remote management

Improved Reliability

- Redundant power & drives available
- Thermally efficient & optimized
- Reduced data center costs



Easy Deployment

- Fully configured & tested
- Pre-cabled, pre-racked
- Quick ship

Learn more at **www.rackable.com/dcpower** or call 1.866.548.2527.
Also look for us at **LinuxWorld 2004** in New York City, January 20-23, Booth #315.

intel inside XEON

IN WITH THE NEW

Rackable® systems

©2003 Rackable Systems. All rights reserved. Intel, Intel Xeon, Intel Inside, and the Intel Inside logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Running `make install` installs the program in /usr/local/games. In that directory, you may find two versions of the program, one called construo.x11 and the other called construo.glut. On the SuSE 8.2 notebook where I tested *Construo*, I found the graphics to be sharper and the text cleaner when using the X11 version (Figure 2).

When the program starts for the first time, you are presented with a clean slate. At the top of the program window, you see some keyboard shortcuts. To the right and left are control buttons. If you abandon a design halfway when you quit, *Construo* returns you to that design the next time you play. If, like François, the idea of starting clean intimidates you, click the Load button and navigate to the Examples directory. There, you are rewarded with a number of prebuilt structures you can modify until you feel more confident to create from scratch.

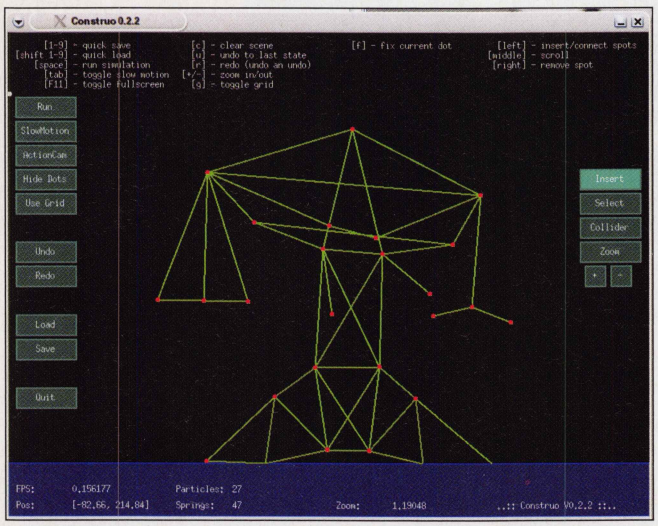


Figure 2. *Construo* makes designing improbable structures fun.

To add a dot, left-click on the screen, move to where you want the opposite point (or joint) to reside and left-click again. To remove a dot, right-click. At some point, you should feel you've created something reasonably stable. Click the Run button to set the laws of physics into motion, and watch how your structure behaves under stress. In all likelihood, your first creation will bend, twist, topple and eventually collapse—which can be a lot of fun. *Construo* lets you set objects in motion (check out the basketball demo), put barriers in the way (colliders) and otherwise release your creative genius.

To modify and strengthen your now—demolished creation, click Undo and your whatever-it-is (or was) returns to its former glory. When you get bored with creating and are ready to start anew, press C to clear the scene, and you once again begin your mechanical ascent to new heights.

Building things on solid ground is interesting enough, but try building them in a 3-D, zero gravity, virtual world. That's the idea behind one of the strangest games this Linux chef has ever run across. The program, called *Ensemblist*, was created for a 2003 Paris game-coding exhibit. This is another one of those deceptively simple ideas that turns out to be something much greater. All you have to do is build the shape presented to you on-screen using simple geometric shapes. This assembly uses simple programming and mathematical principles: boolean, union and intersection.

Have I neglected to mention that the shapes are three-dimensional and floating in a networked virtual space where they can be rotated along any axis? This is something the authors, rixed and dom, call CSG, or constructive solid geometry (Figure 3).

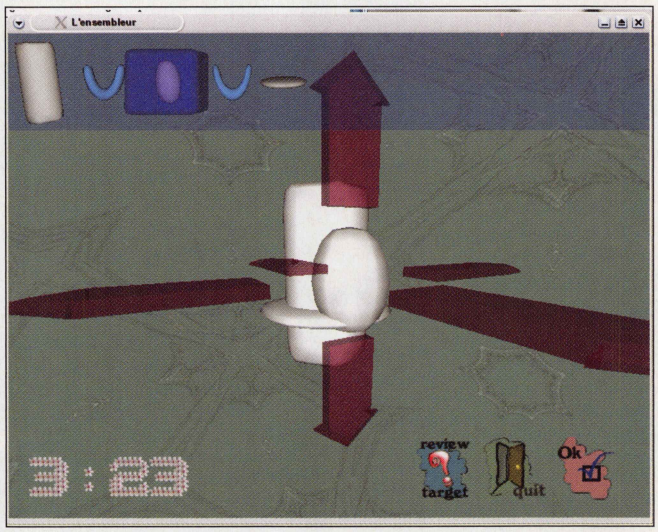


Figure 3. Building Complex Objects from Geometric Primitives in *Ensemblist*'s CSG

Ensemblist is distributed as source from www.nongnu.org/ensemblist. After extracting the file with `tar -xzvf ensemblist.tgz` and switching to the source directory with `cd ensemblist`, you simply run `make`. When the compile finishes, type `su -c 'make install'`. Now, run the program by typing `ensemblist` at the command line. *Ensemblist* installs in /usr/local/games by default.

What follows is a bit strange the first time around. The program connects to the Ensemblist Web site, after which you find yourself floating above a flowchart diagram (Figure 4) where you see your user name and your current score (which is 1). Below that intro box are two options, OK and Quit. With your mouse, you can zoom in and out from this landscape by moving the cursor up or down. Click OK and you are transported to the game options, where you can choose free play or select a campaign that takes you through some predefined levels. You probably want to start with a campaign. You also may want to ask François to fill up your wineglass, to steady yourself.

When the game starts, your selected object is presented in the center of a spherical space bounded by fractal walls. Click the Play icon to the lower right and the action begins. On the bottom left-hand corner, a counter ticks away. The objects you need to use to create the final shape are geometric primitives, and they all are sitting at the top left. These are spaced with union operators. Clicking on the union operators selects between union (looks like a U), intersection (an upside-down U) and minus (a minus sign). At the bottom right, new icons appear so that you can review the original shape, quit or claim victory.

That's all there is to it. Move shapes in and out of the central build area according to these simple rules. This also is where you discover how difficult simple can be. *Ensemblist* is very strange and strangely addictive. On subsequent plays, you can use the --no-net option when starting the game, because the existing levels have been downloaded to your system.

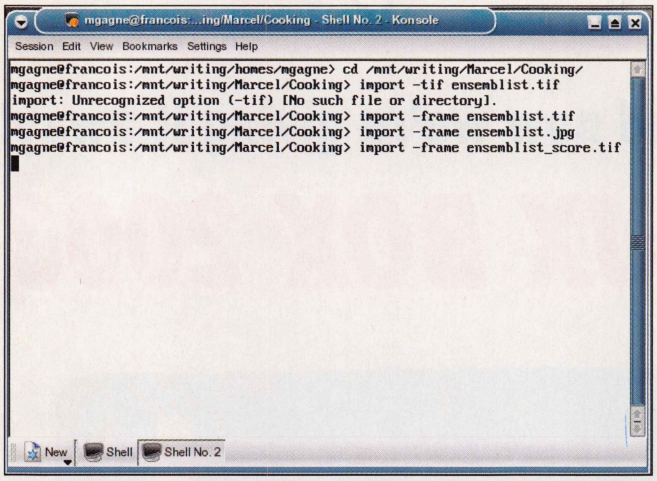


Figure 4. Play Begins in *Ensemblist*'s Surreal Virtual World

Before you click Play next time, take note of the box labeled Editor. Yes, if you feel up to it, you even can create your own levels and help build the *Ensemblist* universe.

Mon Dieu! It seems that we have managed to come to the end of another evening. Perhaps, inspired by the complex wonders that can result from the application of simple ideas, one of you fine people might turn your talents to building a machine that will give us more time, *non*? Stretching time certainly would qualify as taking scalability to new heights. Still, there certainly is time for another glass of wine. Relax, finish what you are working on, and François will pour you a final glass. If you are feeling a little too relaxed, it is nice to know that the only height you need to raise your wine to is your lips. A little joke, *mes amis*. Enjoy! Until next time, *mes amis*, let us all drink to one another's health. *A vôtre santé! Bon appétit!*

Marcel Gagné (mggagne@salmar.com) lives in Mississauga, Ontario. He is the author of the newly published *Moving to Linux: Kiss the Blue Screen of Death Goodbye!* (ISBN 0-321-15998-5) from Addison Wesley. His first book is the highly acclaimed *Linux System Administration: A User's Guide* (ISBN 0-201-71934-7). In real life, he is president of Salmar Consulting, Inc., a systems integration and network consulting firm.

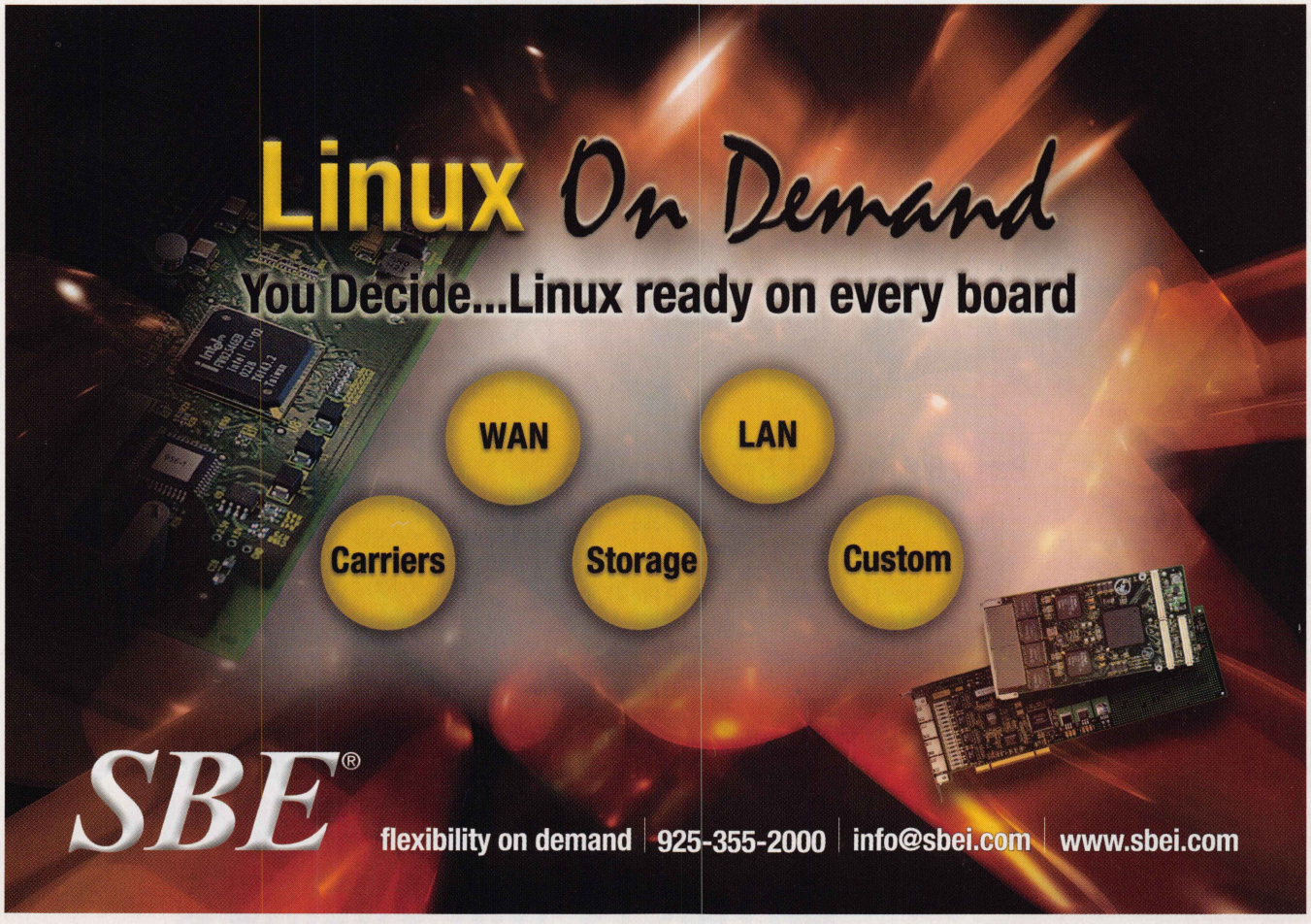
RESOURCES

Construo: www.nongnu.org/construo

Ensemblist: www.nongnu.org/ensemblist

KAtomic: games.kde.org

Marcel's Wine Page: www.marcelgagne.com/wine.html



Monarch Computer Systems®

We built the ULB 2003, and now we're giving it away!

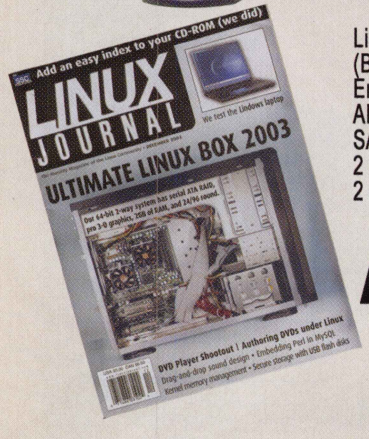
ULTIMATE LINUX BOX 2003

Part # 80761 - This System Features

Lian-Li/Monarch PC 6070B Ultra Quiet Case
(Black Aluminum - Monarch Exclusive)
Enermax 465P-VE-24P Whisper 460 Watt PS
ARIMA HDAMB Opteron WS Board AGP Pro 4X/8X
SATA/IEEE Motherboard
2 AMD™ Opteron™ 246 Processors (Dual CPUs - 2GHz)
2 GB DDR 2700 REG ECC Corsair Memory

16x DVD-ROM
52x32x52 CD-RW
Mitsumi 1.44 MB FDD
3Ware 8500 RAID Controller
RAID 5 Setup
NVidia Quadro FX 1000 AGP 8X 128 MB Dual DVI

$5000 as configured



AS SEEN IN THE DECEMBER 2003 ISSUE OF LINUX JOURNAL

ENTER TO WIN ONLINE @

www.monarchcomputer.com/ulb2k3

What would you do...

...with the most powerful 64-bit Linux workstation you can build?

YOU CAN WIN THIS $5000 PC

QUICK QUOTES on Systems @ www.monarchcomputer.com or call 1-800-611-0875

AMD 64 Opteron

Boost performance on existing Linux applications without recompiling or recoding—so you can get more work done in a shorter period of time, improving overall productivity, and budget efficiency.

ENTER TO WIN ONLINE @ www.monarchcomputer.com/ulb2k3

CONTEST OPEN UNTIL JAN. 31, 2004

Secure Mail with LDAP and IMAP, Part II

An IMAP mail server with an LDAP directory makes things simple, secure and easy for the user. Now Mick explains the tricky parts to make you the company e-mail guru. **BY MICK BAUER**

In the first part of this series on using LDAP with the Cyrus IMAP mail delivery server (*LJ*, November 2003), we got as far as installing and setting up Cyrus IMAP and Cyrus SASL. In this article, we add some users to Cyrus IMAP and configure Postfix to deliver mail to the Cyrus IMAP server.

Cyrus IMAP Documentation

Before we dive back in to Cyrus IMAP configuration and administration, a note about documentation. Cyrus IMAP comes with an administrator's manual in HTML format. In the SuSE distribution, the manual is in /usr/share/doc/packages/cyrus-imapd/doc, and in Simon Matter's Red Hat SRPM distribution (see Part I of this article) it's in /usr/share/doc/cyrus-imapd-2.1.12. The link misleadingly labeled Installation actually leads not only to Cyrus installation instructions but to configuration and administration instructions as well. Besides this documentation, several man pages also are included with Cyrus IMAP, most notably imapd.conf(5), imapd(8) and cyradm(1).

In addition to Cyrus IMAP's included documentation, I recommend the book *Managing IMAP* by Dianna and Kevin Mullet (O'Reilly & Associates, 2000). As far as I know, it's the only book dedicated to IMAP. Although its coverage of Cyrus IMAP doesn't extend to LDAP, it's a well-written book that clearly explains IMAP concepts and Cyrus IMAP administration; it also covers UW-IMAP in some detail.

Using cyradm

Cyrus IMAP comes with a Perl script, cyradmn, that provides the most convenient way to create and manage user mailboxes. You should understand several things before using cyradm. First, you should run cyradm from any account with which you also read e-mail. In other words, you never should use an IMAP administrative account as an e-mail account. Due to unusual write-access permissions, using such accounts to read or send e-mail can have strange negative effects on your server. As we learned last time, Cyrus administrative accounts are named according to the variable admins in /etc/imapd.conf.

Second, cyradm uses the same authentication method as does the rest of Cyrus IMAP. In my previous column, we determined this by setting /etc/imapd.conf's variable sasl_pwcheck_method to saslauthd and by editing /etc/sysconfig/saslauthd to use either LDAP or, in the case of SuSE, to use PAM. PAM itself can be configured to use LDAP for IMAP transactions in the files /etc/pam.d/imap and /etc/openldap/ldap.conf. In short, cyradm identifies and authenticates administrative users with LDAP, assuming you've correctly configured LDAP support in Cyrus IMAP, as described last time.

Finally, to authenticate, cyradm performs an LDAP auth lookup against your user name and password, using the LDAP attribute UID as the search criterion. For each user account you want to allow to run cyradm, therefore, the LDAP record needs to contain definitions for both UID and userPassword. UID is a required attribute and userPassword is an allowed attribute in the posixAccount Object Class, so all IMAP user accounts need to be associated with posixAccount.

This last point has another important ramification: in your OpenLDAP server's /etc/openldap/slapd.conf file, you need to have access control list (ACL) statements granting auth access to the userPassword attribute for whatever LDAP user your IMAP server (or its saslauthd process) uses to bind to the LDAP server (that is, to perform authentications). LDAP ACL statements are described in the slapd.conf(5) man page and in my article "Authenticate with LDAP, Part III" (*LJ*, September 2003).

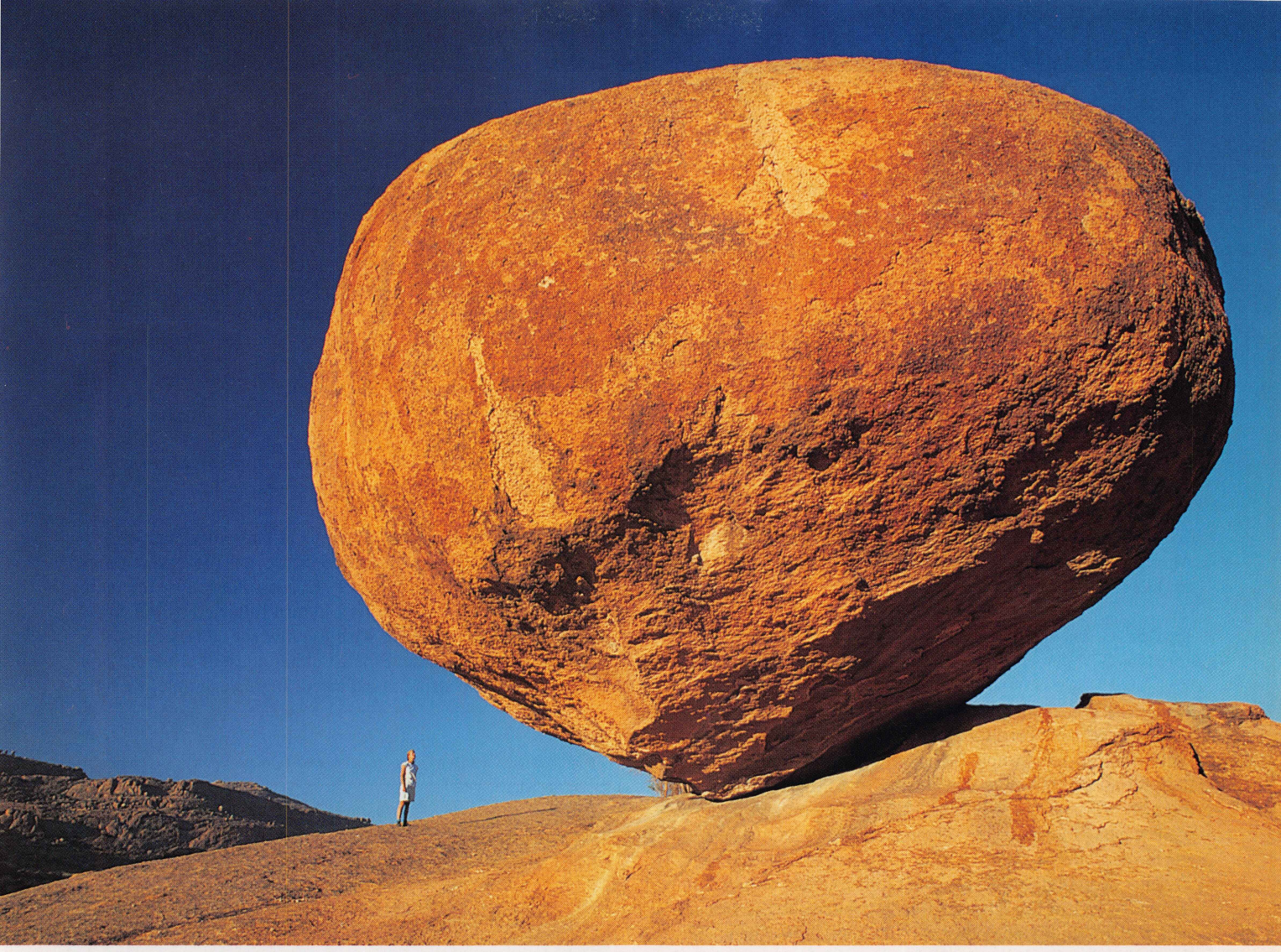
cyradm usually is run as an administrative shell rather than as a command, per se. When you invoke cyradm, supplying your user name plus the host you wish to administer, it prompts you for a password. On successful authentication, it begins an interactive session with its own commands and help screen. cyradm also can be run non-interactively; see the cyradm(1) man page for information on using cyradm for scripting.

The simplest invocation of cyradm is:

```
bash-$> cyradm --user username hostname
```

If you're running cyradm on the same host on which Cyrus IMAP is running, you can use the hostname localhost. If the server you want to administer is a remote host, however, specify its hostname or IP address. By default, cyradm attempts to connect to it over TCP port 143. Because Cyrus IMAP uses this port for clear-text communication, use the --port option to specify TCP port 993 for TLS-encrypted communications instead, like this: `--port 993`. Personally, in such situations I find it simplest to connect to my remote IMAP servers with SSH and then run cyradm locally on the remote host using my SSH session.

Suppose I want to run cyradm locally on my IMAP server and my admin account is called mick_admin. The command



Network backup seems insurmountable if you don't have the right solution.

Introducing Arkeia 5.2 The right solution.

If you're responsible for protecting your company's heterogeneous network, the Arkeia 5.2 Linux or Unix backup server solution is just what you've been looking for.

Arkeia has always been noted for powerful, scalable, reliable and cost-effective backup solutions. Now we've added even more options to give you greater control and ease of use.

New features, greater benefits
Our completely redesigned reporting functions allow you to produce highly customizable machine-based reports as well as more developed messages that inform you about successful operation completion.

Quantum DLTSage™ support for SDLT tape devices helps you predict and prevent storage system errors.

An optimized catalog algorithm lets you dramatically increase backup speed for large file servers—five times faster for 10,000 files within a single directory and up to 50 times faster for 150,000 files!

We've also expanded GUI capabilities for maximized security. You can easily clone tapes for off-site storage. Combine this with Arkeia's Disaster Recovery module, and you can automatically rebuild servers from scratch if they are destroyed or stolen.

In addition, a new hot backup plug-in is available for MS-Exchange that complements our online backup solutions for Oracle, Lotus, DB2 and MySQL.

Try Arkeia 5.2 for 30 days. Free!
The best way to prove that Arkeia 5.2 is the right solution for you is to let you try it—**FREE**—for 30 days. We'll even include free installation tech support. Simply download the demo version at www.arkeia.com.

arkeia®
ENTERPRISE NETWORK BACKUP

www.arkeia.com

would look like this:

```
bash-$ cyradm -u mick_admin localhost
IMAP Password: **********

   localhost>
```

Notice the localhost> prompt after a successful login: I'm now logged in to a cyradm shell session. To see a complete list of available commands, all I need to do is type ? or `help`. There are 20 commands in all, and each can be abbreviated, sometimes in two different ways. The help screen lists all versions of each command.

Creating Mailboxes with cyradm

To create a mailbox, I can use the command `createmailbox`. Alternatively, I can use the abbreviation `create` or even a simple `cm`, like this:

```
localhost> cm user.bwooster
localhost>
```

This is the very model of command-line efficiency, but notice that the user name corresponding to our new mailbox isn't really user.bwooster—it's simply bwooster. The user. prefix must be used for all mailboxes you create in Cyrus IMAP. Thus, to create a mailbox for the user bubba, I'd use the command `cm user.bubba`. To then create subdirectories for that mailbox, I'd use `cm user.bubba.sent`, `cm user.bubba.drafts` and so forth.

This user. prefix is visible only to Cyrus and to its administrators. In fact, when our user Bubba connects to the server with Evolution or some other IMAP client, rather than user.bubba he simply sees a folder named Inbox, even though its real name is user.bubba. Similarly, submailboxes appear as sent, drafts and so on, indented beneath Inbox.

Another thing worth noting about the e-mail account creation command is the lack of any feedback whatsoever from Cyrus upon successful completion. If you're like me, you find this unnerving; you periodically want to use the `listmailbox` command, `lm` for short, to see what you have:

```
localhost> lm
   user.bwooster (\HasNoChildren)
```

Believe it or not, this is all we need to do with Cyrus IMAP to allow user bwooster to receive and read his e-mail, assuming there's an LDAP record with a UID of bwooster. In Cyrus IMAP, creating a new user mailbox has the effect of creating that user's IMAP account. But before I move on to the topic of configuring the Postfix MTA to deliver e-mail to Cyrus IMAP, a few words about Cyrus IMAP ACLs.

Cyrus IMAP ACLs

Each mailbox in a Cyrus IMAP system can have one or more ACLs associated with it in which each ACL defines which actions a given user may perform on the referenced mailbox or folder. By default, a new mailbox has only one ACL, one that grants the mailbox's owner full administrative rights over the mailbox.

Interestingly, administrators by default have only lookup and administer rights on the new mailbox. You can look up the name of the mailbox using the `listmailbox` command, and you can set ACLs on it. But if you need to delete the mailbox, you first must create an ACL for the mailbox that grants your administrative account administrative rights. This is a feature, not a bug; it helps prevent things from getting deleted accidentally.

Continuing with our example, below are the commands for removing the mailbox we just created, using our administrative account mick_admin:

```
$ cyradm -u mick_admin localhost
IMAP Password: *****

localhost> setaclmailbox user.bwooster mick_admin all
localhost> deletemailbox user.bwooster
```

The second command issued here is of particular note; it begins with the cyradm command `setaclmailbox`, which may be abbreviated as `sam` or `setacl`. This is followed by the mailbox in question (user.bwooster), in turn followed by the account name to which we wish to grant (or deny) access, mick_admin in this case. Finally, either a group of permission codes or a special string is indicated. In this example, we have

Table 1. cyradm ACL Permission Codes (adapted from the cyradm(1) man page)

Permission	Description
l	Lookup (visible to LIST/LSUB/UNSEEN)
r	Read (SELECT, CHECK, FETCH, PARTIAL, SEARCH, COPY source)
s	Seen (STORE \SEEN)
w	Write flags other than \SEEN and \DELETED
i	Insert (APPEND, COPY destination)
p	Post (send mail to mailbox)
c	Create and delete mailbox (CREATE new submailboxes, RENAME or DELETE mailbox)
d	Delete (STORE \DELETED, EXPUNGE)
a	Administer (SETACL)
none	Special string meaning no permissions
read	Special string meaning lrs
post	Special string meaning lrsp
append	Special string meaning lrsip
write	Special string meaning lrswipcd
all	Special string meaning lrswipcda

More Specials on the Web 01.01.04

lantechservers.com
866.306.5590

Lantech helped me plant the seed in my farm...render farm that is...
— Oliver M.

5-Years Warranty
All systems are tested and preloaded with your choice of Linux Distribution

AMD 64 Opteron | intel inside XEON

LS-1UX300.i

1U IDE Solution

1U Lantech Rackmount Chassis
Single Intel Pentium 3.0GHz - 800MHz
2GB ECC Registered RAM(4GB max)
2 x 120GB Hard Drives with 8MB CACH
1 x 20GB OS DRIVE
-RAID 0/1 for striping
2 x 10/100 ethernet adapters
52X CD-ROM Drive (optional)
1.44 MB floppy disc drive (optional)

Featured at **$1,949**

LANTECHSERVERS manufactures secure, reliable, innovative, and cpmpetitively priced server tailored to YOUR needs and demands rather then ours. We provide massive computing solutions to various fields of intrests such as *Internet DataCenters nationaly, ISP's, Scientific Research facilities, 3D Render Farms, Video/Multimedia Streaming, CAD designers, Video Editing and rendering farms,* and the list goes on and on...**Botton Line, we gurantee your satisfaction! Our specialists are standing by to assist you. Call us today.**

Bring Pluto Miles Closer to Earth

LS-1UX28b.i

1U HIGH DATA YIELD IDE

1U Lantech Rackmount Chassis
Dual Intel Xeon 2.8GHz - 512k 533HMz
2GB ECC Registered RAM (8GB max)
2 x 200GB Hard Drives 7200RPM 8MB
1 x 20GB OS DRIVE 7200RPM
-RAID 0/1 for striping
2 x 10/100/1000 ethernet adapters
52X CD-ROM Drive (Optional)
1.44 MB floppy disc drive (optional)

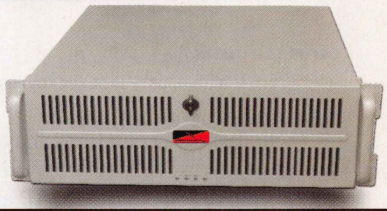
Featured at **$2,595**

LS-2UX26c.s

2U IDE RAID DATA SERVER

2U Lantech Rackmount Chassis
Dual Intel Xeon 2.6GHz - 512k - 800MHz
2GB ECC Registered RAM(8GB max)
4 x 200GB Hard Drives with 8MB CACH
1 x 40GB OS DRIVE
-RAID 0/1/5/10
2 x 10/100/1000 ethernet adapters
52X CD-ROM Drive (Optional)
1.44 MB floppy disc drive (Optional)

Featured at **$3,195**



LS-4UOP260.i

4U IDE Massive Storage

2U Lantech Rackmount Chassis
Dual Intel Xeon 3.0GHz 512k - 533MHz
2GB ECC Registered RAM(6GB max)
8 x 200GB SATA Hard Drives 8MB
-RAID 0/1/5/10
2 x 60GB OS DRIVE (RAID 0/1)
2 x 10/100 ethernet adapters
52X CD-ROM Drive (Optional)
1.44 MB floppy disc drive (Optional)

Featured at **$5,295**

SO-9001 Certified

Lantechservers.com is not responsible for any typographical errors. Pricing and availability are subject to change without notice.

the special string `all`, which is short for all permissions. To delete the user.bwooster mailbox, it would have been sufficient simply to specify c, for create or delete mailbox or submailboxes. Other possible ACL permissions are listed in Table 1.

ACLs are covered in detail in the cyradm(1) man page and are explained in Cyrus IMAP's HTML documentation. I highly recommend that you get into the habit of at least reviewing, if not always customizing, the ACLs on each mailbox you create with cyradm. On some sites, it may not be necessary for users to retain the default permission c. If all user submailboxes (user.whomever.sent, user.whomever.saved and so on) are created for them by you, for example, you may prefer that they not have the ability to create new ones or delete them accidentally.

Configuring Postfix to Deliver to Cyrus IMAP

In Part I, I described the role of mail delivery agents (MDAs) as delivering mail to mailboxes. Being an MDA, Cyrus IMAP can deliver mail, but it first must receive that mail from a message transfer agent (MTA). The most popular MTA is Sendmail, but a simpler and more secure option is Wietse Venema's excellent Postfix (www.postfix.org). As Postfix is my MTA of choice, and because it's available either as the default MTA or as an option in most major Linux distributions nowadays, it's the one I explain in detail here.

Does your IMAP server need to reside on your organization's SMTP relay? It can, but it doesn't have to. It may make more sense from the standpoints of security and performance to keep your SMTP relay dedicated to that purpose. You then can have your IMAP server run its own instance of Postfix that receives mail from the dedicated SMTP relay rather than directly from other networks' MTAs. In either case, we assume the MTA from which IMAP receives its mail is running on the same host as Cyrus IMAP.

Three files need to be edited in order to configure Postfix to transfer mail to Cyrus. The first file is /etc/postfix/main.cf, in which we need to add or uncomment this line:

```
mailbox_transport = cyrus
```

The second file we need to edit is /etc/postfix/master.cf, in which we need to add or uncomment these two lines:

```
cyrus     unix  -       n       n       -       -       pipe
user=cyrus argv=/usr/libexec/cyrus/deliver -r ${sender} ${user}
```

Actually, the second line may differ on your system; the syntax of Cyrus' deliver program has changed over the years. If you installed both Cyrus IMAP and Postfix from your Linux distribution's current CDs or download site, the included /etc/postfix/master.cf file should work without tweaking. If you installed either Cyrus IMAP or Postfix from source code, however, you may need to do some tweaking and Googling to get the second line exactly right. One key piece of the second line is the path in argv=/usr/libexec/cyrus/deliver, which must point to your local system's Cyrus deliver command.

The third and final Postfix file to edit is /etc/aliases; you may keep yours in /etc/postfix/aliases. Unless you're using LDAP for alias lookups—a process too involved for this article, but which I describe in the Sidebar—you need to have at least one entry in aliases for each Cyrus mailbox, plus any aliases to those mailboxes you need. For our example user Bubba, /etc/aliases needs the line:

```
bubba:    bubba
```

Simple enough, right? We omit the user. prefix; Cyrus mailboxes are referred to by user name. If your Cyrus (LDAP) user names correspond to local system user names, you don't need aliases entries for those users. But part of Cyrus' attraction lies in its not requiring users to have shell accounts.

If Bubba is our organization's marketing analyst, we also can add the line:

```
marketing_weasel: bubba
```

After you edit your aliases file, don't forget to use the postalias command to generate a new alias database:

```
bash-$> postalias hash:/etc/aliases
```

POSTFIX AND LDAP

In this article I describe how to use LDAP to authenticate Cyrus IMAP users, but I cover Postfix only so far as pointing Postfix mail delivery at Cyrus. In fact, Postfix also has LDAP functionality: it can use LDAP to resolve e-mail aliases to mailbox names.

That is, you can configure Postfix to check both its local /etc/postfix/aliases database for e-mail-alias-to-mailbox-name mappings and also to query the local LDAP service or a remote one. This can save considerable administration time; rather than maintaining separate alias and user databases, you can do it all in LDAP.

However, Postfix on Red Hat 7.3 (and possibly on later versions) doesn't have LDAP support compiled in. To determine whether your version of your distribution of choice has LDAP support compiled in its Postfix package, use the command `postconf -m`. If LDAP isn't listed among the supported Postfix modules, you need to uninstall your Postfix package and build it yourself from source.

See www.postfix.org for more information and for Postfix source code. Be sure to read the instructions in ./README/LDAP_README in the Postfix source code, which explains how to compile in Postfix's LDAP functionality—the default Postfix Makefile does not do so automatically. Be sure also to read the file /etc/postfix/samples/sample-ldap.cf, which contains the parameters you need to add and configure in /etc/postfix/main.cf to get LDAP alias lookups working. The latter step is extremely important, and it may take some tinkering to get it working properly.

If you forego all this and choose instead to maintain Postfix's aliases file separately (the old-fashioned way), then don't worry; using or not using LDAP with Postfix has no ramifications whatsoever on Postfix's ability to interact with your LDAP-authenticated Cyrus IMAP software.

Conclusion

This is not all you need to know in order to be a Cyrus IMAP administrator, but hopefully it's enough to get started in building an LDAP-enabled Cyrus IMAP server. With the topics we've covered or touched on in these two articles, you now can go on to advanced topics, including how to let users change their LDAP passwords; how to let users use the LDAP server as an address book; how to set up shared IMAP folders securely; and how to set up a secure Web mail interface, such as SquirrelMail for Cyrus IMAP.

Mick Bauer, CISSP, is *Linux Journal*'s security editor and an IS security consultant in Minneapolis, Minnesota. He's the author of *Building Secure Servers With Linux* (O'Reilly & Associates, 2002).

RESOURCES

Cyrus IMAP Home Page (source, documentation and so on): asg.web.cmu.edu/cyrus/imapd.

Mullet, Dianna, and Kevin Mullet. *Managing IMAP*. Sebastopol, California: O'Reilly & Associates, 2000.

"Secure Mail with LDAP and IMAP, Part I": www.linuxjournal.com/article/6998.

TECH TIPS

To remove spurious Control-M characters, or carriage returns, from MS-DOS format text files, do this:

```
tr -d '\r' < dirtyfile > cleanfile
```

If you're writing a game in Perl and you want the random numbers to be really random, seed the pseudo-random number generator from the kernel's real random number generator, like this:

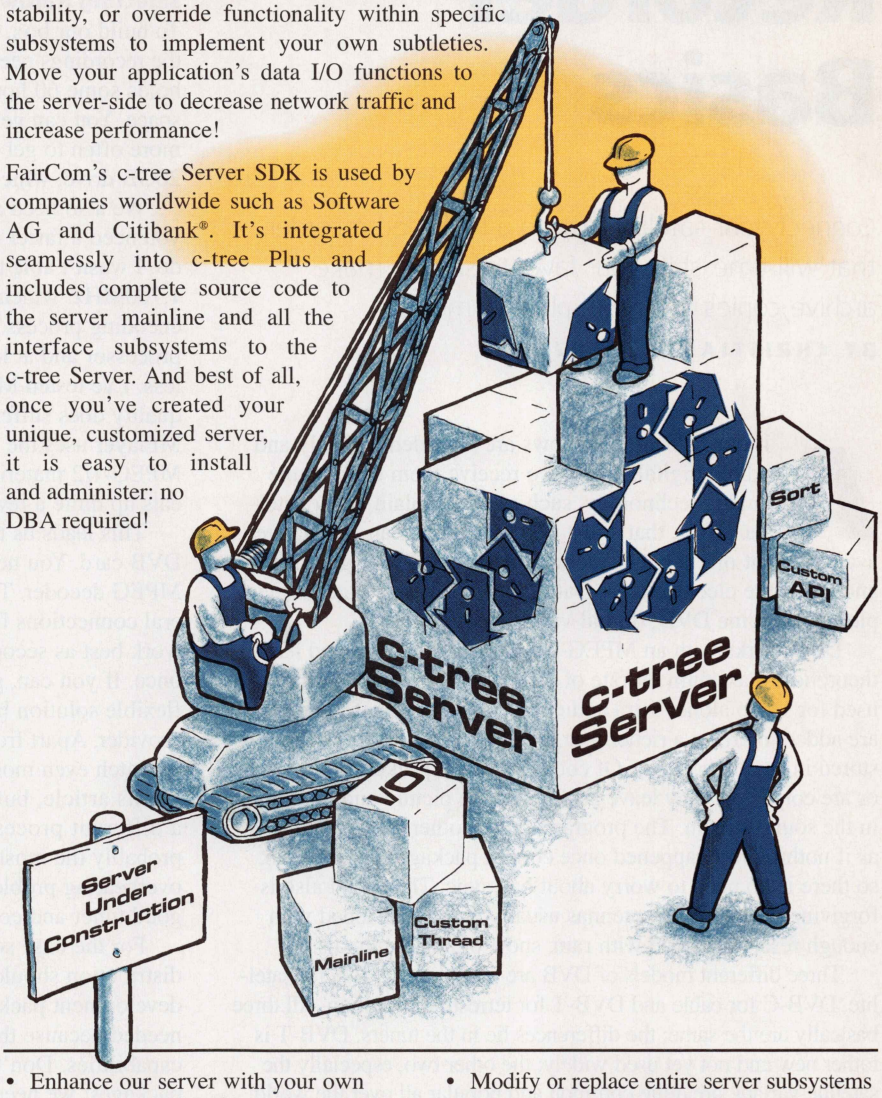
```
BEGIN: {
  my $random;
  open (RANDOM, "/dev/random")
    or die "Can't open " .
           "/dev/random: $! ";
  read (RANDOM, $random, 4);
  close RANDOM;
  srand(unpack('L', $random));
}
```


CUSTOMIZE
YOUR DATABASE SERVER WITH THE C-TREE® SERVER SDK

Today's database demands are often too complex for traditional database servers. The functionality and precise level of control you need is simply not available. Perhaps you need alternate sort criteria for your data or a special twist in the threading or communication logic.

FairCom's c-tree® Server SDK allows you to create a customized, industrial-strength server designed for your particular needs. Use FairCom's kernel, with over 20 years of proven stability, or override functionality within specific subsystems to implement your own subtleties. Move your application's data I/O functions to the server-side to decrease network traffic and increase performance!

FairCom's c-tree Server SDK is used by companies worldwide such as Software AG and Citibank®. It's integrated seamlessly into c-tree Plus and includes complete source code to the server mainline and all the interface subsystems to the c-tree Server. And best of all, once you've created your unique, customized server, it is easy to install and administer: no DBA required!



- Enhance our server with your own custom server-side functionality
- Move functionality from the client-side to the server-side to reduce network traffic and increase performance
- Modify or replace entire server subsystems
- Complete source for the server mainline, key server subsystems, and client-side
- Flexible OEM licensing

Visit www.faircom.com/ep/lj/sdk today to take control of your server!

FairCom®
www.faircom.com

USA	573.445.6833
EUROPE	+39.035.773.464
JAPAN	+81.59.229.7504
BRAZIL	+55.11.3872.9802

DBMS Since 1979 • 800.234.8180 • info@faircom.com

Other company and product names are registered trademarks or trademarks of their respective owners. © 2002 FairCom Corporation

Personal Video Recorder Basics

Connect your satellite dish to a PC-based recorder that will time-shift your favorite shows, make archive copies and, with plugins, more.

BY CHRISTIAN A. HERZOG

All our favorite TV shows are recorded digitally and edited digitally. But, we receive them using some analog technology, such as cable, plain air or satellite. Now, that's just wrong. Luckily, an alternative is available that offers better picture quality, Dolby Digital sound and EPG, the electronic program guide. Ladies and gentlemen, please welcome DVB, digital video broadcasting.

DVB works with an MPEG-2 compressed stream and has a theoretical maximum bit rate of 15Mbit/s. Because DVB never is used for video alone, extra audio tracks and other information are added to create a richer user experience. All of this data is stored in small packages. Of course, if some packets don't arrive or are corrupted, they leave artifacts in the picture and possibly in the sound stream. The program, on the other hand, continues as if nothing has happened once correct packages arrive again, so there is no need to worry about bad sync. The signal also is forgiving, because the antennas usually are dimensioned with enough reserve to cope with rain, snow or small animals.

Three different models of DVB are available, DVB-S for satellite, DVB-C for cable and DVB-T for terrestrial reception. All three basically are the same; the differences lie in the tuners. DVB-T is rather new and not yet used widely; the other two, especially the satellite variant, are quite common and popular all over the world.

Building the VDR

Modern set-top boxes available from major manufacturers have some nice features, such as hard disk recording and MP3 playback. Nevertheless, they lack more advanced options, including archival of recordings to SVCD or some MPEG-4 sibling. With the advent of affordable DVD burners and media, backing up your favorite TV shows to DVD is well within reach. In Europe, digital personal video recorders (PVRs) carry a hefty price tag of around 500 EUR, and most come without a hard disk. A full-featured DVB-S card, on the other hand, can be found at on-line retailers for as low as 165 EUR.

But, what good is cheap hardware without good software? The program VDR, video disc recorder, enables you to build a powerful set-top box on your own using your favorite flavor of Linux and a DVB card. The machine I built incorporates basic features, such as watching TV, recording and time-shifting, plus advanced features, including MP3/Ogg playback, playback of all video formats supported by MPlayer and backup of the recorded material to MPEG-4, video CD or DVD. A commercial set-top box hardly stands a chance against this feature list.

Selecting Hardware

To build our box, we need some hardware. Bear in mind that the recordings need a lot of space. A 120GB hard drive typically holds some 60 hours of video, which should be plenty of space. You can get away with less if you back up your movies more often to get them off the drive, but I recommend at least a 20GB drive, which holds about three or four movies.

We also need a processor. If you want to encode the videos, you need a faster one; if not, an old 200MHz machine should do. I wasn't able to find anything slower than a Celeron 1,700MHz, which is more than enough power, even for the encoding process. Playback using MPlayer also requires a fast processor and at least 1GHz, though it's rumored to work with less. I've tested MPlayer on a slower machine, and the image quality does suffer quite a bit. The reason for this is the way MPlayer uses the MPEG-2 decoder on the DVB board. Non-MPEG-1/2 material is converted to MPEG on the fly, which eats up quite a few processor cycles.

This leads us to the most important piece of hardware; the DVB card. You need a full-featured card with a hardware MPEG decoder. These are more expensive, but they have several connections for sound and TV. Cards like the WinTV Nova work best as secondary cards to record several programs at once. If you can, go for the satellite option. It is by far the most flexible solution because you are not dependent on some cable provider. Apart from that you can link up several satellite dishes to watch even more channels. I also discuss the DVB-S variant in this article, but deploying a different solution is not really a different process. I opted for a Hauppauge Nexus-s. It's probably the most expensive card, but it doesn't suffer from the overheating problems older models experienced, plus it has a good tuner and comes with an infrared remote and receiver.

For the base software load, I used Red Hat Linux 9, but any distribution should do. A small installation with GCC and development packages for libjpeg should be enough. X isn't needed because the full-featured DVB cards have video-out capabilities. Don't forget to install all the kernel development packages; we need those to compile the DVB driver.

Installing Drivers

Once the base distribution is up and running, we need a driver for the card. You can get the CVS version at linvdr.org/download/vdr/Developer. At the time of this writing, linux-dvb.2003-09-05.tar.bz2 is the latest version. The current drivers sometimes hang when disconnecting the satellite cable or the reception drops to zero. You then have to remove and re-insert the drivers, which does not always work, leaving you with the need to reboot to get it up and running again. These hangs can be especially pesky if you're simply recording something or are in the middle of a movie, but they usually don't happen.

Now go to /usr/src, unpack the driver snapshot and `mv` it to DVB. Renaming the directory is important because certain

patches and plugins rely on directory names. Go to the DVB directory and type `make` to compile the driver and some useful applications that help you scan the satellites to retrieve a list of channels. `make install` is not needed because the runvdr script we use later takes care of the module loading. It's important to run the makedev.napi script after compiling the drivers, as this script creates the needed entries in /dev.

Scanning for Channels

If you live outside of Europe or don't use the Astra satellites, you have to use a different channel list. Scanning for channels is an automated process. A tool called scan comes with the DVB driver, and you can find it in the /apps/scan directory. Invoke it with the `-o vdr` option so the output file is in VDR's channel format. To capture the newly created channel file, you need to redirect the programs standard output with this command:

```
./scan -o vdr > channels.conf
```

Installing VDR

Fetch vdr-1.2.5.tar.bz2, unpack it in /usr/src and change to that directory. Installing VDR can be a bit tricky. Because you can use some patches to spice up the features, you quickly can end up in patch hell if you're not careful. The best idea, if you want to use multiple patches, is to get your hands on an all-in-one patch. I used nothing but the Elchi patch, which gives the rather dull default VDR interface a nice face-lift. If you have the right patch for your VDR version, you shouldn't encounter any problems.

The range of additional functions added by plugins goes from simple games to e-mail alerts to full-featured DVD playback. Here, install only two, the remote plugin and the MP3/MPlayer playback plugin. The remote plugin is necessary only when using the original remote from Hauppauge. The MP3/MPlayer plugin, on the other hand, is a must-have.

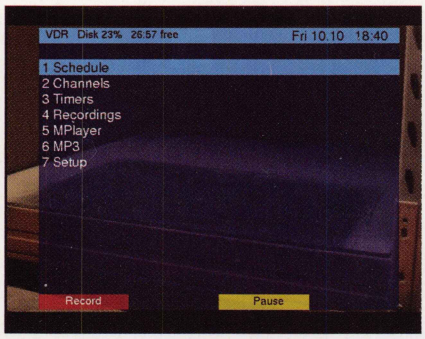


Figure 1. VDR Main Menu with the Elchi Patch Applied

Change to /usr/src/vdr-1.2.5/PLUGINS/src and unpack the two packages. VDR's Makefile won't build the plugins until you strip the version information from the directory names, so rename them to remote and MP3. The MP3 plugin has some additional requirements, namely libsndfile, libmad and libid3tag. Since Red Hat ships without MP3 support, you have to install them manually. Fetch them from www.freshrpms.org, and don't forget to install the development packages. When everything is set, type `make REMOTE=plugin NEWSTRUCT=1 all plugins`. The REMOTE=plugin parameter adds another input method using the remote plugin. You can use Lirc to select whatever remote you happen to find; VCR ones do rather well. Simply add `REMOTE=lirc` in that case. Keyboard support is enabled by default and shouldn't be disabled, as it can be very helpful for debugging. The `NEWSTRUCT=1` is needed to tell the plugins to search for the new drivers in /usr/src/DVB.

Having compiled everything, we now need to edit the start-up script a bit. As a basis, use the one supplied with the remote plugin. You can find it in the misc directory, named runvdr.remote. This script loads a keymap so the signals from the remote are decoded. This is used only with Hauppauge's IR receiver. If you don't have one, use the runvdr script in VDR's root as a starting point. Move the runvdr.remote script to VDR's root and fire up your favorite editor. On line 24 you should find the parameters by which vdr is started. Mine looks like this:

```
VDRCMD="$VDRPRG -w 60 -P scanner \
        -P\"mplayer -M /video/plugins/mplayer.sh\" \
        -P mp3 \
        -P\"remote -i /dev/input/event1\" $*"
```


Wireless Connectivity
- Cellular Modem
- 802.11b WiFi

TS-5600 586 Single Board Computer

Linux 2.4

$279 qty 1

Shown with optional IDE Compact Flash, A/D, DAC, RS-485 and Orinoco WiFi adapter.

- PC compatible: BIOS, DOS, RTC, I/O & memory map
- User programmable PLD for digital I/O
- Low power +5v @ 900mA
- 4.3" x 5.6"
- TS-Linux 3.0 is open source GNU/Linux - Glibc 2.2.5
- Apache, SSH, Perl, bash, FTP, kernel drivers

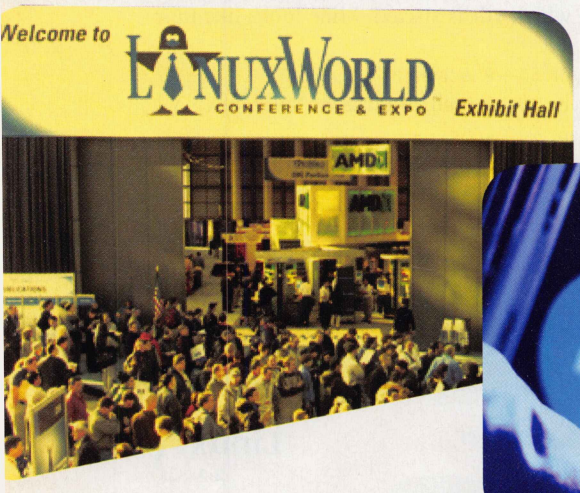
- 133 MHz x86
- PCMCIA slot
- 10/100 Ethernet
- PC/104 Bus
- Compact Flash
- 32-64 MB SDRAM
- 6 ch 12-bit A/D
- 2 ch 12-bit DAC
- 23 Digital I/O
- 2 Serial Ports

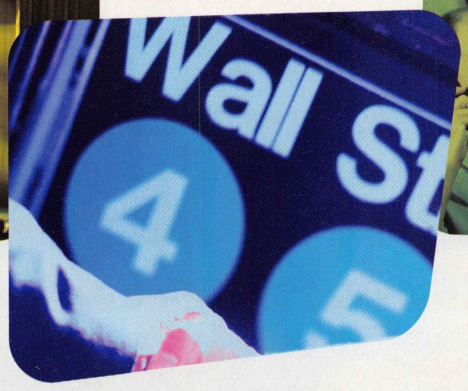
Technologic SYSTEMS

(480)-837-5200
www.embedded586.com

LINUXWORLD®
CONFERENCE & EXPO

Where Open Minds Meet





Keynote Speakers

Chris Stone
Vice Chairman, Office of the CEO, Novell, Inc.
Wednesday, January 21
9:15 am – 10:00 am

Dave Dargo
Vice President, Linux Program Office, Oracle
Wednesday, January 21
11:45 am – 12:30 pm

Tom Killalea
Vice President of Infrastructure, Amazon.com
Wednesday, January 21
2:45 pm – 3:30 pm

Sam Greenblatt
Senior Vice President and Chief Architect, Computer Associates Intl.
Thursday, January 22
11:30 am – 12:15 am

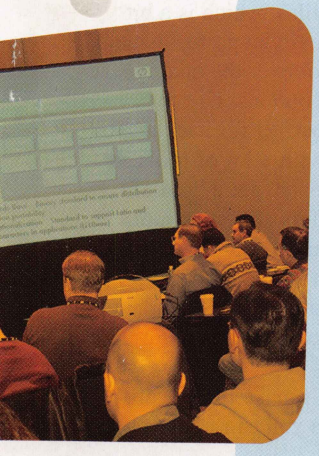
Ross A. Mauri
General Manager, e-Business on demand, IBM Systems Group
Thursday, January 22
1:45 pm – 2:30 pm

CONFERENCE: January 20 – 23, 2004

EXPO: January 21 – 23, 2004

THE JAVITS CENTER: New York, NY

LinuxWorld Conference & Expo is the world's leading and most comprehensive event focusing on Linux and open source solutions. This January, discover how companies across the globe have achieved higher profits and increased their productivity by utilizing Linux, the fastest growing operating system in the world.



- Get in-depth coverage on the latest Linux and open source developments and learn about innovative product solutions from the industry's top companies.
- Network with the leaders in the open source movement and discuss with your peers how to best leverage the technology for your organization.
- Participate in LinuxWorld's world-class education program and benefit from interactive training in the all-new Hands-on Labs!
- Attend the 2nd annual Linux Financial Summit and hear how Wall Street firms are leveraging Linux to achieve a lower total cost of ownership. (Sponsored by BEA)
- Discover real world practical solutions and find answers to today's most critical IT issues.

Cornerstone Sponsor

ORACLE

ca Computer Associates

IBM intel hp invent Novell

Sun microsystems

suse simply change

REGISTER ONLINE WITH REFERENCE CODE: A-LJJ

www.linuxworldexpo.com

IDG WORLD EXPO

If you're unsure what you need to add, simply run `vdr -help`, and you should see a list of all usable modules and their options, along with all of VDR's options. Don't worry, it's fairly easy to find out what to add. To finish up, we need a base directory; the default (as defined in Make.config) is /video. This directory holds all the recordings and configuration files. Copy the sources.conf and your channels.conf to /video.

The only thing left for us to do is write configuration files for the MPlayer/MP3 plugin. Start by creating a directory named /video/plugins. Now we need two files, one called mp3sources.conf and one called mplayersources.conf. Writing them is simple. For starters, add something like `/video/music;Local files;0` for mp3sources.conf and `/video/compressed;Local files;0` for mplayersources.conf and save them. To get MPlayer to work with the DVB card, you have to recompile. Grab a copy from www.mplayerhq.hu, and add `--with-extraincdir=/usr/src/DVB/include` to your configure options. Let the configure script run, recompile and install, and you're good to go.

As you can see from the runvdr script, the MPlayer plugin uses a special shell script to start MPlayer, mplayer.sh. You can get that from batleth.sapienti-sat.org/projects/VDR. The package consists of only two files, mplayer.sh itself and mplayer.sh.conf, which hold some configuration options. If your machine is too slow to play back files using MPlayer, you should try tweaking the settings in this file.

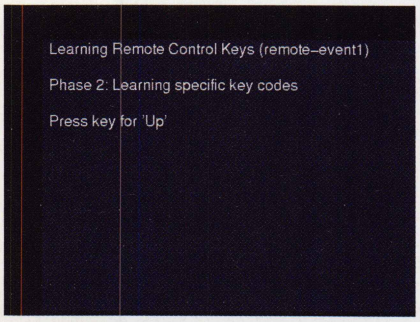


Figure 2. Defining the Keys for Your Remote

Roll Film and Back It Up!

Go back to /usr/src/vdr-1.2.5 and run `runvdr.remote`. If you use Red Hat, set the environment variable LD_ASSUME_KERNEL=2.4.1, because VDR doesn't yet work with the native posix layer that Red Hat introduced in the latest version. The modules for the DVB card then are loaded, and the VDR is started. Hook up your TV, and you should see a black screen prompting you to define the keys on your remote. After finishing the wizard, you're ready to watch TV, record shows and remove commercials. You can listen to your MP3s and watch videos. There's a manual in VDR's root that explains how to record and edit TV events, using the time-shift feature.

Back It Up

In case you're disappointed that the end of the article is within reach, don't worry; there still are some optional things you can do. The automatic backup feature has some limitations. Although the (S)VCD backup works flawlessly, the DivX encoding does not crop the picture to remove black bands, should they exist. This has quite a negative impact on bit rate, size and overall picture quality. If you really want a high-quality, small-size MPEG-4, you should back it up manually. The improved picture quality is well worth the trouble.

Figure 3. The information bar shows the program name, running TV show and what's on next.

VDR splits its recordings into 2GB files, which is a bit inconvenient for transcoding the videos. If you go for manual conversion, which gives you finer control over the quality/size aspect, mencoder or transcode are good options. Use the speedy mencoder, which I found to be perfect for backups to MPEG-4, or transcode, which comes with a lot of tools. If you favor the I-don't-want-to-care approach, get a hold of VDRCONVERT. The README file offers a pretty simple approach to installing it, and at least you can watch some TV while downloading and compiling. With VDRCONVERT you have to change some scripts and configuration files to adapt the DVD/(S)VCD resolutions to NTSC, in case PAL is not used where you live.

It's too bad that a Linux PVR doesn't make the TV programs themselves any better, but I guess you can't have everything, can you?

Christian A. Herzog is a programmer focused on Web development using open-source technologies. He's still on his never-ending quest to bring a Linux-based device to every home and company he comes across. Write him at noeffred@gmx.net.

RESOURCES

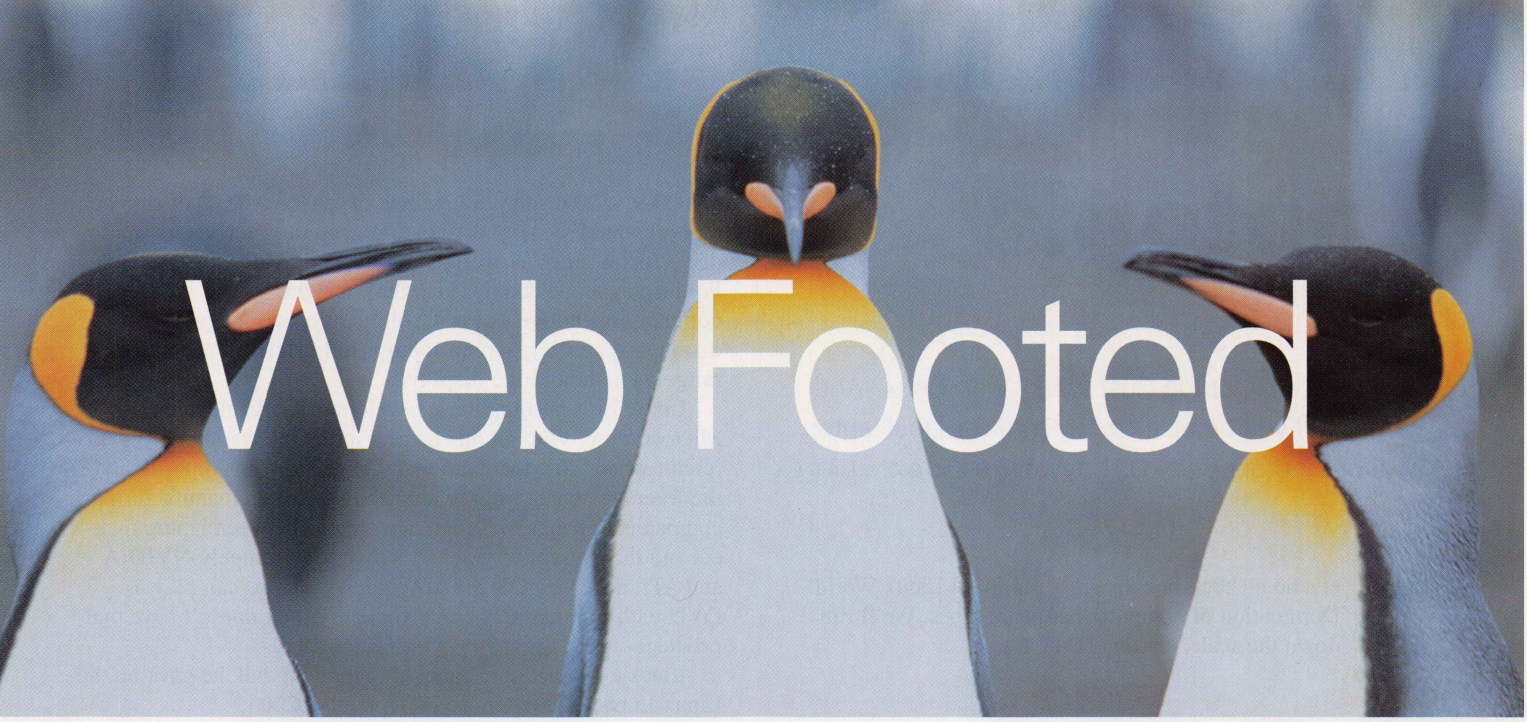
Libraries: www.freshrpms.org

LIRC: www.lirc.org

MPlayer: www.mplayer.hu

mplayer.sh: batleth.sapienti-sat.org/projects/VDR

Plugins, Scripts and Patches for VDR: www.vdrportal.de/board/portal_downloads.php?site=6



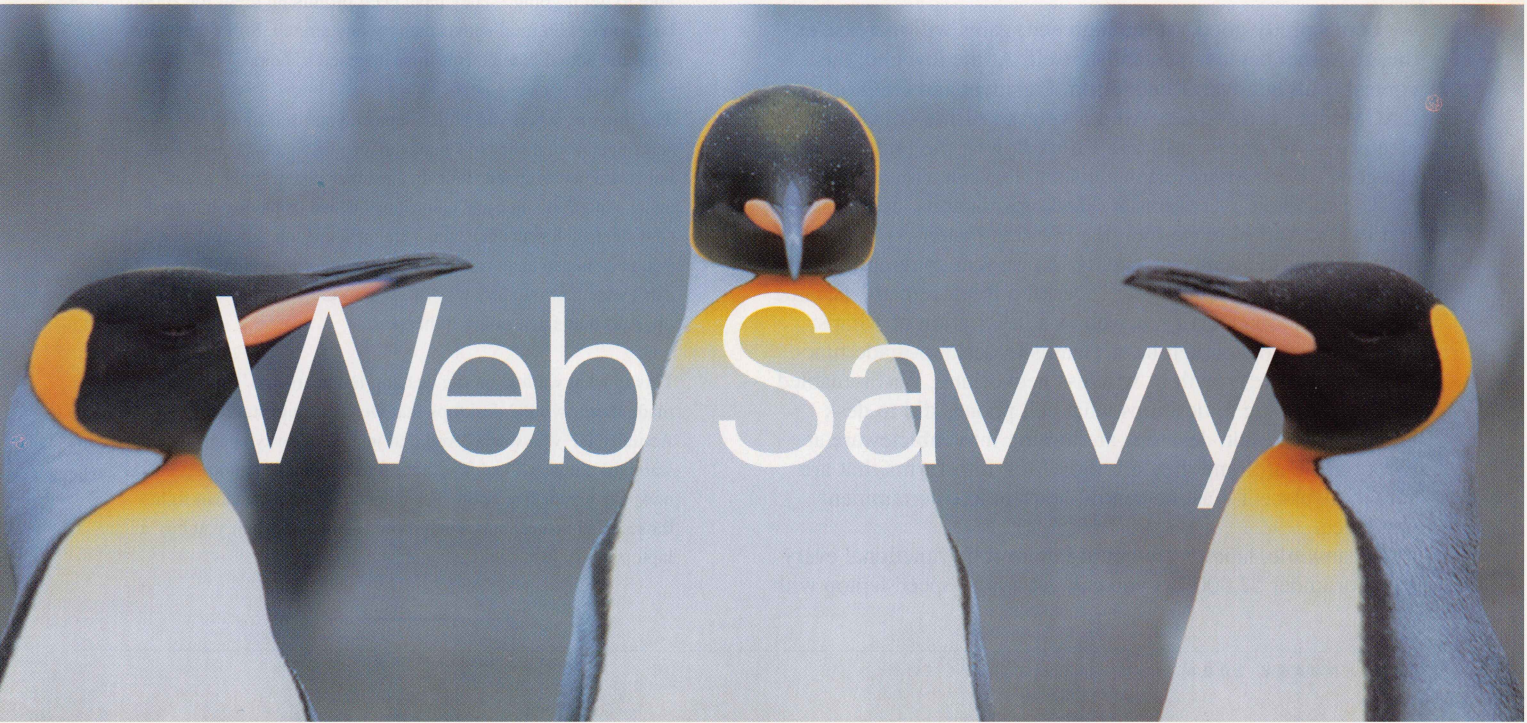
The right management software allows you to seamlessly integrate Linux into your infrastructure.

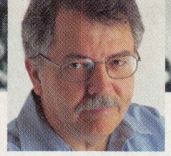
Linux Management Solutions

No matter what type of Linux implementation you have planned, you'll still need management software to ensure that it's seamlessly integrated across your enterprise. Our management software solutions help maximize the potential of Linux across your entire business by letting you take full advantage of its unique scalability and open-source programming. As a result, you can increase productivity while decreasing total cost of ownership. To find out more about our continuing commitment to Linux innovations, go to ca.com/linux today.

ca Computer Associates®

© 2003 Computer Associates International, Inc. (CA). All rights reserved.





Laptopia

Will your next laptop "just work" with Linux? And when will you be able to buy a major-label laptop without a proprietary OS anyway? **BY DOC SEARLS**

Let's do a check-box progress report on Linux World Domination of computer hardware types. We'll run down the scale, starting at the top:

- Mainframes
- Big enterprise database machines
- Big compute farms
- Symmetrical multiprocessing machines
- Small- and medium-sized business machines
- Engineering workstations
- Desktops
- Laptops
- Notebooks
- PDAs
- Embedded devices

Looks pretty good. At the high end, gains will increase as the 2.6 kernel begins to establish itself. At the low end, the contest is all but over. Ken Nelson, President and CEO of CACMedia (cacmedia.tv) and a veteran of the consumer electronics business, tells me Linux "owns the embedded category". Even Microsoft is a distant also-ran.

The standout exceptions are desktops, laptops, notebooks and PDAs. And of those four, the one that matters is laptops. That's where the center of gravity in personal computing is moving today. In the last year, "Desktop Replacement" became a leading laptop category as well as a description of what's happening to the marketplace. Thanks to desktop-grade functionality and the growth of wireless networking, Wi-Fi-enabled laptops are doing to desktops what cell phones did to desk phones. As a result, desktops are turning into cubicleware at work, while at home they are morphing into forms that are sure to displace TV as the central household entertainment appliance.

Meanwhile, laptops are getting more fully functional every day. For under $2,000 US you can get a high-speed laptop with a 1600 × 1200 display, a gigabyte of RAM, 80+GB of storage, a good sound system, a first-rate graphics subsystem and a drive for burning as well as playing both CDs and DVDs. For display alone, laptops kick desktop butt. As one hardware company executive put it, "You want a two megapixel display—get a big-screen laptop. It's priced in the same range as a standalone display."

Linux isn't pushing these developments; Apple and Microsoft are. With all due respect to the innovations involved (credit where due: there are many, especially on Apple's part), the leverage comes from relationships between manufacturing companies. Thanks to design agreements between laptop makers and the sources of graphics subsystems, namely NVIDIA and ATI, new Mac OS X and Windows laptops can play a DVD with minimal CPU involvement. Linux doesn't have that privilege—yet.

Linus explained the situation in the long talk he gave on the Linux Lunacy Geek Cruise in September 2003:

> Quite honestly, none of the laptop vendors support Linux at all, really. To be real. Some of them go through a certain amount of motions. They support Linux in certain configurations if it's not too painful. But the amount of support tends to be okay (to a limited degree). It may not suspend. It may not actually do half of the things you want a laptop to do. But you *can* run Linux on it. I expect that to change as companies start to use Linux more on the desktop. If you have a few big companies that just say, "Hey, Linux *has* to work on our laptops", suddenly hardware manufacturers will start caring a whole lot more....
>
> So, the good news is laptops are moving away from the embedded machine kind of thing. They are getting *so* standardized. Especially with chipsets like Centrino. If you use the Pentium-M and you don't use Centrino, you are doing stuff wrong—except for the fact that they don't support (802.11) A&G; and right now you can't get Centrino drivers for Linux....
>
> I've also heard that they exist, but other people at Intel say "That's crap. We have it on the road map, but we haven't been able to get it going." They have been promising them for 2004, but I am not an Intel spokesman by any stretch. So I don't know what the actual date is. But it is supposed to come.
>
> The thing is, when you built a laptop, you used to have to scrounge around people's backyards to find strange pieces of hardware just to make it all fit. And that is definitely going away. And that's not just Centrino. Instead of having hundreds of different chipsets that you wire up a million different ways, you're going to have maybe five different chipsets, and you can't wire them up any way other than the way they are wired up. And that's just going to happen.
>
> This is what we had on the desktop ten years ago. Compaq made their own PC desktops that weren't quite standard. Actually HP was worse. And that just went away because of standard chipsets. And it's starting to happen in laptop space now. So, a year from now, I'd expect...assuming we can fight those ACPI issues...it's much more likely that when you buy a laptop it will just work.

Markets get made in exactly two ways. The first is "find customer needs and satisfy them." When a bunch of companies start doing that and competing to make the best products and services, you have a market. Reduced to a single phrase, Marketing 101 teaches "Necessity is the mother of invention." The second way is less obvious, but without it we wouldn't have a single technology category. It goes like this: make something so new and cool that customers think they have to have it. They hardly teach this in marketing school, but if they did, the single-phrase version would be "Invention is the mother of necessity."

When people ask you how Apple can drive whole markets and establish new technology standards (hard-case floppies, SCSI, flat screens, USB, FireWire, GUIs, Wi-Fi and so on) over and over again, point them to that second marketing principle. Apple does a great job of inventing needs. It's a skill that goes beyond simple innovation. Steve Jobs' genius is not his obsession with art; that's a real asset, but it's also a red herring. His genius is knowing how invention can lever the world.

Invention is not what Microsoft does. And it's not what Linux does, either. Microsoft innovates, as they'll tell you, over and over again—and they're right. The company comes up with an endless series of modest but marketable improvements to goods largely invented elsewhere. Although Microsoft has an abundance of red herring assets (credit where due), its real genius may lie in the very hole where we currently find Linux in respect to laptops: in relationships with hardware OEMs. These relationships are highly muscular on Microsoft's part and deeply involved. Look at any laptop today, and you see a little sticker that says "designed for Windows XP" or something similar, along with a Microsoft Windows logo. You can imagine the extent to which Microsoft labors, within its hardware OEM relationships, to make sure Linux has the hardest possible time running on laptops.

And don't discount the manufacturers' shared interest in the same resistance. No tier-one hardware manufacturer wants to see the world filled with white box or build-it-yourself laptops. What makes each of their laptops unique may be the very thing that keeps Linux from running on every model across the board, from one manufacturer to another. But that's not a problem. Not in the long run, which may not take very long at all.

For display alone, laptops kick desktop butt.

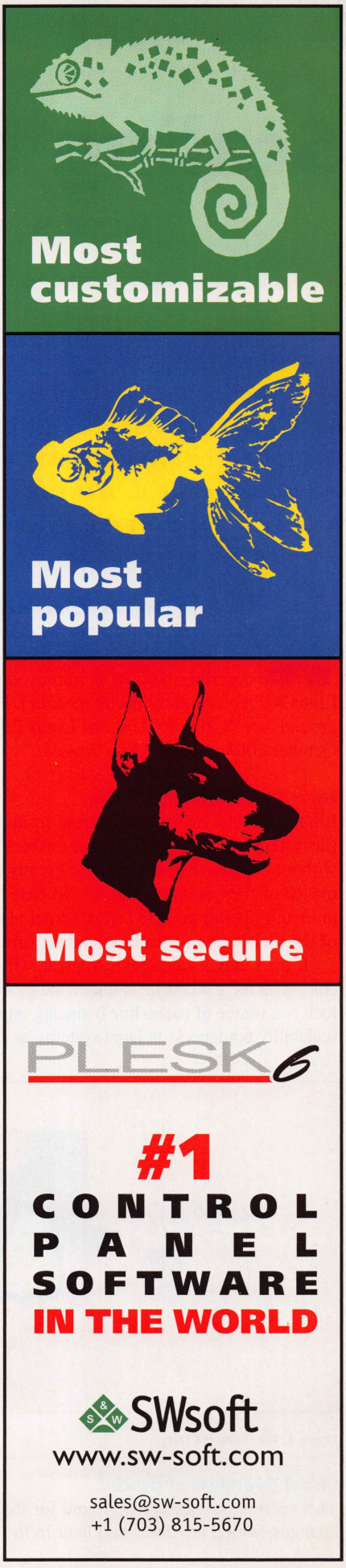
In the emerging laptop market ecology, three parties will represent three different forces: Apple the *inventor*, Microsoft the *innovator* and Linux the *disrupter*.

In *The Innovator's Dilemma: When New Technologies Cause Great Companies to Fail*, Clayton Christensen lays out the last two roles rather clearly. Innovators find it hard to disrupt themselves. They resist the inevitable, even as they see it coming. Incumbent innovators could be found in every one of the computing categories we checked at the start of this column. Linux disrupted all of them. The ones that embraced the disruption, starting with IBM, have made the most of innovation as well.

Linux also will disrupt laptops and the remaining categories for the simple reason taught in Marketing 101: necessity will mother invention. One of these months, some company—a FedEx, a Boeing, a General Motors, a Siemens—will call on Dell, HP and IBM to compete for filling a gargantuan Linux laptop order. The relationship with Microsoft will be strained for the winner of the contest, but they'll do the deal. They'll lean on Intel to release the Linux device drivers for Centrino. They'll work with Canon and Sony to get the device drivers written for the cameras, camcorders and scanners. They'll finish hammering out the ACPI issues. And we'll have good, cheap Linux laptops being marketed, with real advertising, by the big hardware OEMs.

Then you can start checking off the rest of the list.

Doc Searls is senior editor of *Linux Journal*.



Scaling dcache with RCU

Reorganizing the way Linux caches filename lookups is a big win for helping to scale to large servers. BY PAUL E. MCKENNEY, DIPANKAR SARMA AND MANEESH SONI

RCU (read-copy update), a synchronization technique optimized for read-mostly data structures, recently was added to the Linux 2.6 kernel. This article describes how RCU improved the scalability of Linux's directory-entry cache (dcache). For some more background, see "Using RCU in the Linux 2.5 Kernel", in the October 2003 issue of *Linux Journal*.

Linux Directory-Entry Cache

Linux's dcache maintains a partial in-memory image of the filesystem hierarchy. This image enables pathname lookup without expensive disk transfers, greatly increasing the performance of filesystem operations. To ease handling of mount and unmount operations, the Linux kernel also maintains an image of the mount tree in struct vfsmount structures.

If the Linux 2.4 dcache is so great, why change it? The difficulty with the 2.4 dcache is it uses the global dcache_lock. This lock is a source of cache-line bouncing on small systems and a scalability bottleneck on large systems, as illustrated in Figure 1.

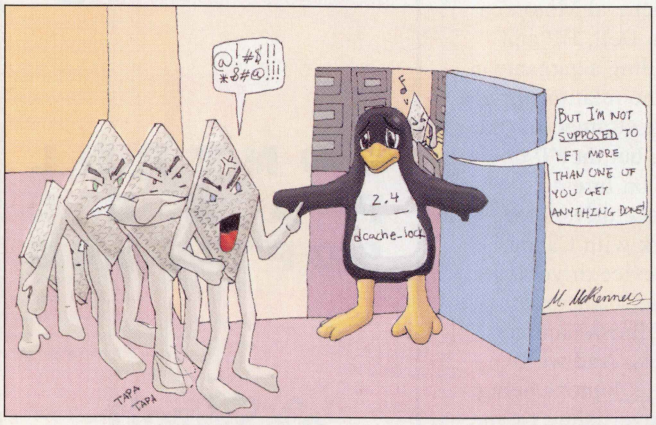


Figure 1. Tux Doing His Duty

Visual Overview of dcache

This section provides background for the RCU-related dcache changes, which are described later in the article. Readers desiring more detail should dive into the source code.

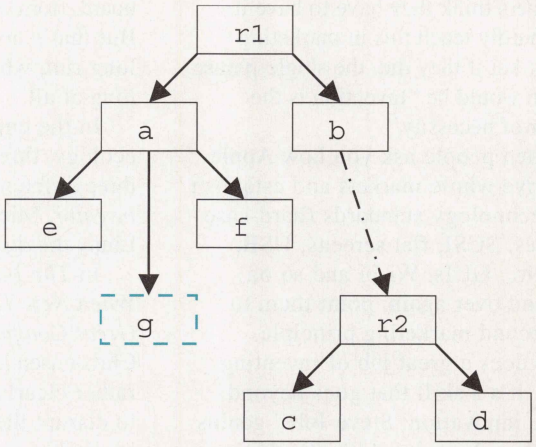


Figure 2. Example Filesystem Tree

This section uses the example filesystem tree shown in Figure 2, which has two mounted filesystems with roots r1 and r2, respectively. The second filesystem is mounted on directory b, as indicated by the dashed arrow. The file g has not been referenced recently and therefore is not present in dcache, as indicated by its dashed blue box.

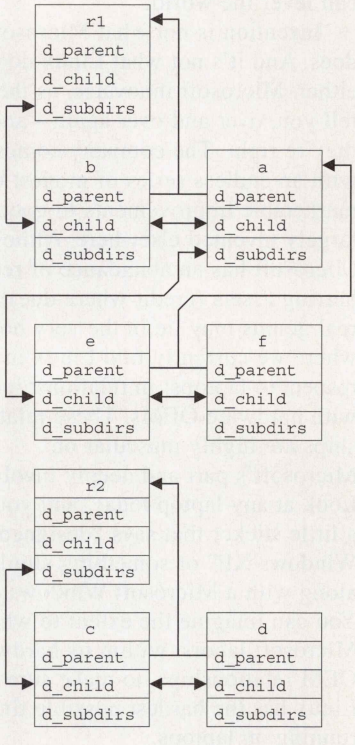


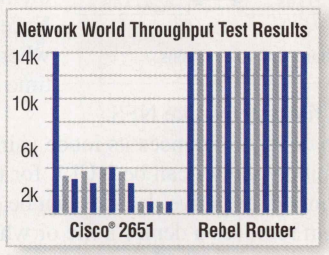
Figure 3. dcache Representation of the Example Filesystem Tree

The dcache subsystem maintains several views of the filesystem trees. Figure 3 shows the directory structure representation. Each dentry representing a directory maintains a doubly linked circular list headed by the d_subdirs field that runs through the child dentries' d_child fields. Each child's d_parent pointer references its parent. The mountpoint (dentry b) does not refer-

THE

REVOLUTION RELOADED

What is router performance? Is router performance defined by the biggest brand, or can you experience it in the real world? It's time to get some answers when you hear a little voice inside asking, **"What did I get for the money?"** Many have paid a high price to own the biggest brand, but can they accept the truth? In a recent head-to-head lab test conducted by Network World Magazine (Vol 20, Issue 28), the industry's largest router manufacturer declined to participate. When asked why, they said, **"... users are interested in things other than performance."** Can you imagine them saying this to their customers? In the past five years, ImageStream has introduced the world's lowest cost routers for OC12/STM4, OC3/STM1, DS3/E3 and high-density T1/E1 applications. Now, independent tests prove again that **ImageStream routers blow away the monster brand** in terms of throughput, latency, and jitter. And if the best price and performance weren't enough, every ImageStream router includes a 31-day performance guarantee, 24/7 technical support, and free software upgrades for life. Linux® has always made the impossible possible. **The choice is yours, the red LED or the blue?**



Independent tests confirm ImageStream's superior performance in terms of throughput, latency and jitter.

ImageStream™
Routers for the Real World

800.813.5123
www.imagestream.com
1.574.935.8484

© Copyright 2003 ImageStream Internet Solutions, Inc. All rights reserved worldwide. ImageStream and "Routers for the Real World" are trademarks of ImageStream Internet Solutions, Inc. Linux is a registered trademark of Linus Torvalds. All other trademarks are property of their respective owners.

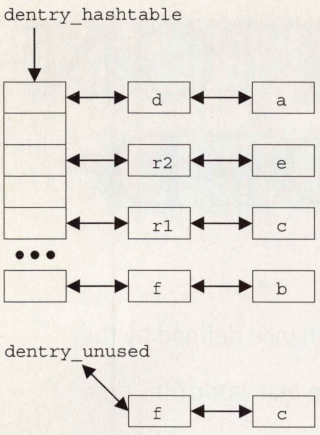


Figure 4. dentry Hash Table

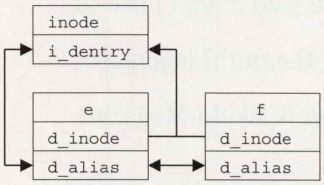


Figure 5. Hard-Link Alias Chains

ence the mounted filesystem directly. Instead, the mountpoint's d_mounted flag is set, and dcache looks up the mounted filesystem in mount_hashtable, a process that is described later.

Although one could search the d_subdirs lists directly, this would be a slow process for large directories. Instead, _ _d_lookup() hashes the parent directory's dentry pointer and the child's name, searching the global dentry_hashtable for the corresponding dentry. This hash table is shown in Figure 4, along with the LRU list headed by dentry_unused. Any dentry in the LRU list usually is in the hash table as well. Exceptions include cases where parent directories time out, such as in distributed filesystems like NFS.

Each dentry references its inode using the d_inode pointer. This d_inode pointer can be NULL for negative dentries, which lack an inode. Negative dentries can be generated when a filesystem removes a dentry's file or when someone tries to lock a non-existent file. Negative dentries can improve system performance by failing repeated accesses to a given non-existent file without having to invoke the underlying filesystem. Similarly, hard links result in multiple dentries sharing an inode, as shown in Figure 5.

Figure 6 shows a high-level dentry state diagram. The normal dentry's life goes as follows:

1. d_alloc() allocates a new dentry for a newly referenced file, leading to state New.

2. d_add() associates the new dentry with its name and inode, leading to state Hashed.

3. When done with the file, d_put() adds the dentry to the LRU list and sets its DCACHE_REFERENCED bit in its d_vfs_flags field, leading to state LRU Ref (Hashed).

4. If the file is referenced again while in the LRU Ref (Hashed) state, dget_locked(), usually called from d_lookup(), marks it in use. If it still is in use at the next prune_dcache() invocation, it is removed from the LRU list, leading again to state Hashed.

5. Otherwise, prune_dcache() eventually removes the DCACHE_REFERENCED bit from the d_vfs_flags field, leading to state LRU (Hashed).

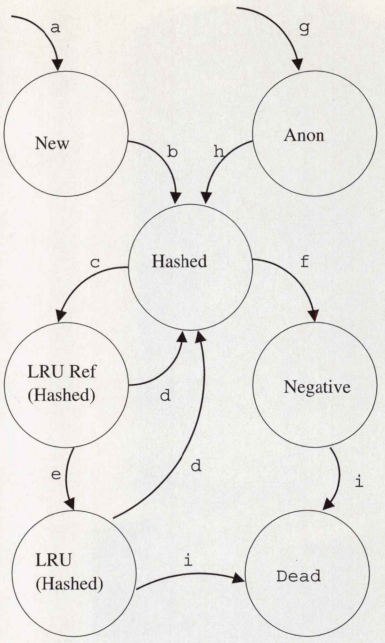
6. As before, if the file is referenced again, dget_locked() marks it in use so that prune_dcache() can remove it from the LRU list, leading again to state Hashed.

7. Otherwise, the second consecutive call to prune_dcache() invokes d_free() via prune_one_dentry(), resulting in state Dead.

Other paths through Figure 6 are possible. For example, if a distributed filesystem converts a cached file handle into a new dentry, it invokes d_alloc_anon() to allocate the dentry when the corresponding object's parent is no longer represented in the dentry cache. Similarly, using d_delete() to delete the file or directory underlying a given dentry would move that dentry to the Negative state. On last close, it would be advanced to "Dead".

Figure 7 shows the mount_hashtable data structure used to map the mountpoint dentry to the struct vfsmount of the mounted filesystem. This mapping hashes the pointer to the mountpoint dentry and the pointer to the struct vfsmount for the filesystem containing the mountpoint. This combination of dentry pointer and struct vfsmount allows multiple mounts on the same mountpoint to be handled more gracefully.

The example filesystem layout shown in Figure 2



a. d_alloc()
b. d_add()
c. dput()
d. dget_locked() / prune_dcache()
e. prune_dcache()
f. d_delete()
g. d_alloc_anon()
h. d_splice_alias()
i. prune_dcache() / d_free()

Figure 6. dentry State Diagram

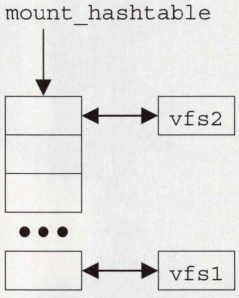


Figure 7. Traversing Mountpoints

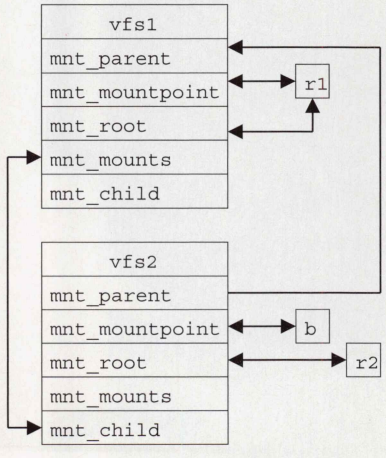
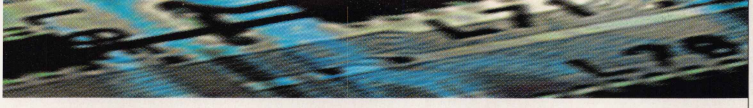


Figure 8. VFS Mount Tree



A full parallelization of dcache would be quite complex and was deemed too risky for the latter part of the 2.5 effort.

would result in struct vfsmount structures as shown in Figure 8. The vfs1 structure references the root dentry r1 both as the mnt_mountpoint and the mnt_root, because this filesystem is the ultimate root of the filesystem tree. The vfs2 structure references dentry b as its mnt_mountpoint and r2 as its mnt_root. Thus, when the mount_hashtable lookup returns a pointer to vfs2, the mnt_root field quickly locates the root of the mounted filesystem.

The overall shape of the mounted filesystems is reflected in the mnt_mount/mnt_child lists. These lists are used by functions such as copy_tree() while doing loopback mount, which need to traverse all the filesystems mounted in a particular subtree of the overall pathname namespace.

Applying RCU to dcache

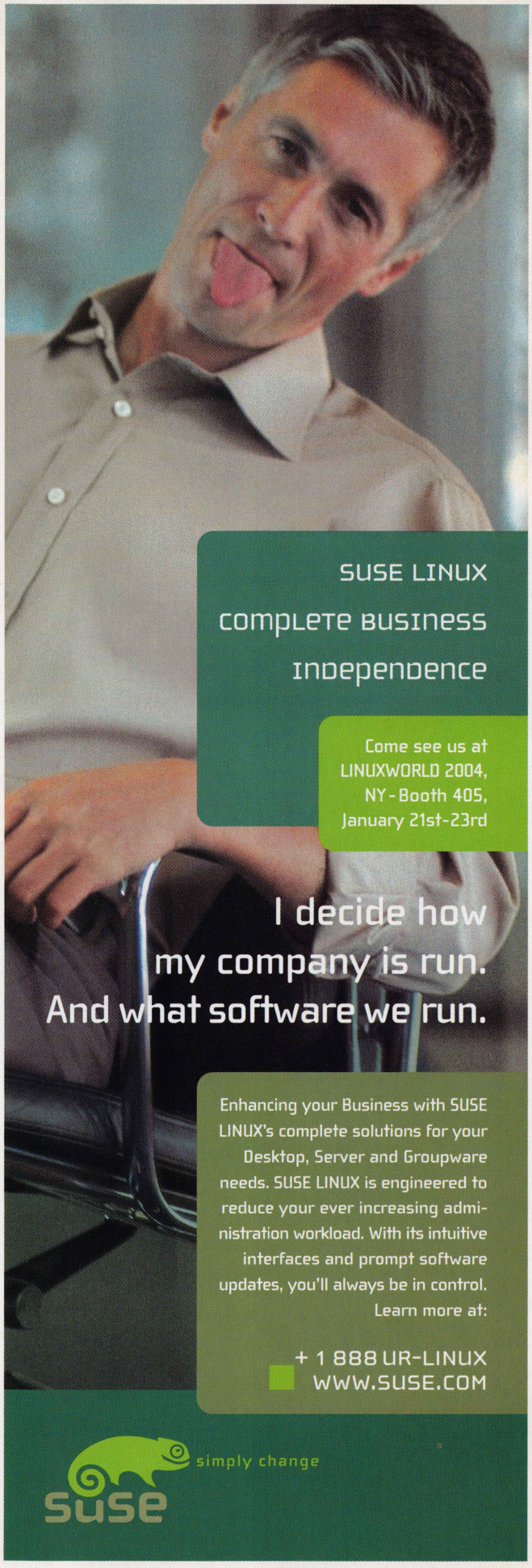
A full parallelization of dcache would be quite complex and was deemed too risky for the latter part of the 2.5 effort. The 2.6 dcache is but one step along the road to RCU; the goal for 2.7 is to walk the entire path without acquiring any locks. Pathname segment lookup is performed by the _ _d_lookup() function shown in Listing 1. The _ _d_lookup() function is invoked with a pointer to the parent directory's dentry and the name to be looked up. The name is passed in a struct qstr, which contains a pointer to the string, its length, a precomputed hash value for the dcache hash table and the name itself, if desired.

Lines 5–7 unmarshall the struct qstr. Line 8 hashes the combination of the name and the parent dentry pointer into the global dcache hash table, yielding a pointer to the corresponding hash chain.

Lines 12 and 56 demark the RCU-protected segment of the code, disabling preemption in CONFIG_PREEMPT kernels, as specified by the Reader-Writer-Lock/RCU analogy described in the article entitled "Using RCU in the Linux 2.5 Kernel" in the October 2003 issue of *Linux Journal*. Lines 13–55 loop through the elements in the selected hash chain, looking for the matching dentry. Line 18 issues a memory barrier but only on DEC Alpha. On other CPUs, the data dependency implied by the pointer dereference suffices, so on these CPUs, line 18 generates no code.

Because this lookup acquired no locks, it may be racing with a rename system call. Such a system call could move a dentry to another hash chain, taking this lookup with it. Lines 21 and 22 check for this race, but they are not sufficient in and of themselves. Therefore, line 23 takes a snapshot of the number of times that the current dentry has been subjected to a rename, the dcache d_move() function, which is used later to determine if any renames raced with the path walk. Line 24 is a memory barrier to ensure that the snapshot is not reordered by either the compiler or the CPU.

Lines 25–28 check the name hash and the parent dentry. If either fail to match, this dentry cannot be the target of our lookup. Lines 29–41 do the full name comparison, with memory barrier for DEC Alpha at line 30. Filesystem-specific name



Listing 1. Lock-Free Pathname Segment Lookup

```
 1 struct dentry *
 2 __d_lookup(struct dentry * parent,
 3             struct qstr * name)
 4 {
 5   unsigned int len = name->len;
 6   unsigned int hash = name->hash;
 7   const unsigned char *str = name->name;
 8   struct hlist_head *head = d_hash(parent,hash);
 9   struct dentry *found = NULL;
10   struct hlist_node *node;
11
12   rcu_read_lock();
13   hlist_for_each (node, head) {
14     struct dentry *dentry;
15     unsigned long move_count;
16     struct qstr * qstr;
17
18     smp_read_barrier_depends();
19     dentry = hlist_entry(node, struct dentry,
20                          d_hash);
21     if (unlikely(dentry->d_bucket != head))
22       break;
23     move_count = dentry->d_move_count;
24     smp_rmb();
25     if (dentry->d_name.hash != hash)
26       continue;
27     if (dentry->d_parent != parent)
28       continue;
29     qstr = dentry->d_qstr;
30     smp_read_barrier_depends();
31     if (parent->d_op &&
32         parent->d_op->d_compare) {
33       if (parent->d_op->d_compare(parent, qstr,
34                                   name))
35         continue;
36     } else {
37       if (qstr->len != len)
38         continue;
39       if (memcmp(qstr->name, str, len))
40         continue;
41     }
42     spin_lock(&dentry->d_lock);
43     /*
44      * If dentry is moved, fail the lookup
45      */
46     if (likely(move_count ==
47                dentry->d_move_count)) {
48       if (!d_unhashed(dentry)) {
49         atomic_inc(&dentry->d_count);
50         found = dentry;
51       }
52     }
53     spin_unlock(&dentry->d_lock);
54     break;
55   }
56   rcu_read_unlock();
57   return found;
58 }
```

comparison functions may be provided, for example, for case-independent filesystems, as shown on line 33.

If execution reaches line 42, we have found a child dentry with a matching name. We then acquire the child dentry's lock on line 42. Because we have a lock on each dentry, the level of contention on these individual locks is much lower than on the original dcache_lock. Nonetheless, life is not perfect. For example, the lock on the root dentry still is subject to contention, a topic discussed later.

The child dentry possibly was renamed after the d_move_count snapshot was acquired on line 23. Therefore, lines 46–47 check the current value of d_move_count against the snapshot. If the check passes, the child dentry has not been renamed out from under the lookup, and lines 48–51 increment a reference count—but only if the entry still is hashed.

Line 53 releases the child dentry's lock, and line 54 breaks out of the hash-chain search loop. Line 57 returns a pointer to the child dentry if the lookup was successful, or NULL otherwise.

A failure of __d_lookup() does not mean that failure is returned to the user process. The file actually may exist, but has not been loaded yet into dcache.

This function does not protect, however, against all rename-race hazards. One additional race is caused by the fact that dcache uses hlist rather than list for the dcache hash chains. It uses hlist to save memory, because hlist requires one rather than two pointers in the list header. This does mean, though, that hlist, unlike list, is not circular. It therefore is possible that a particular dentry will be renamed such that it lands in a previously empty dcache hash chain. If this happened at the right time, the __d_lookup() function could return search failure incorrectly.

An incorrectly returned search failure is handled by the upper-level d_lookup() function, shown in Listing 2. Any racing renames are detected by the read_seqretry() function on line 13. As the problematic case results only in spurious failure, the check is made only on NULL return from __d_lookup().

Listing 2. Pathname Segment Lookup Rename-Race Resolution

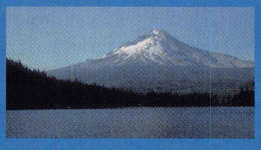
```
 1 struct dentry *
 2 d_lookup(struct dentry * parent,
 3          struct qstr * name)
 4 {
 5   struct dentry * dentry = NULL;
 6   unsigned long seq;
 7
 8   do {
 9     seq = read_seqbegin(&rename_lock);
10     dentry = __d_lookup(parent, name);
11     if (dentry)
12       break;
13   } while (read_seqretry(&rename_lock, seq));
14   return dentry;
15 }
```

Deferred-Free

The d_free() function must defer freeing of a given dentry until a grace period has elapsed, because any number of ongoing path walks might be holding references to that dentry. Deferment is accomplished in the d_free() function shown in

The Compilers and Tools for Scientists and Engineers.

C, C++, F77, F90 and HPF • 32-bit and 64-bit Linux
Optimized for Intel and AMD CPUs • 64-bit support on AMD64
Workstation, Server and Cluster configurations • Fast compile times
Native OpenMP • Native SMP auto-parallelization
Native distributed-memory parallelization • Cache tiling
Function inlining • SSE/SSE2 Vectorization • Loop unrolling
Interprocedural optimization • Profile-feedback optimization
Large file support on 32-bit Linux • 64-bit integers and pointers
F77 pointers • Byte-swapping I/O • VAX and IBM extensions
OpenMP/MPI/threads debugging • OpenMP/MPI/threads profiling
Interoperable with g77/gcc/gdb • PDF and printed documentation
Electronic purchase, download and upgrades • Tech support
Network-floating licenses • Academic and volume discounts



The Portland Group™ Compiler Technology

www.pgroup.com ++01 (503) 682-2806 sales@pgroup.com

The registered trademarks and marks are the property of their respective owners.

STMicroelectronics

Listing 3, where line 5 uses the call_rcu() primitive to defer the destructive actions in the d_callback() function until after a grace period has elapsed. The d_callback() function is shown in Listing 4; it simply frees large names stored separately (lines 5–7), if appropriate, then frees the dentry itself on line 8.

Listing 3. Deferred-Free of dentry Structures

```
1 static void d_free(struct dentry *dentry)
2 {
3   if (dentry->d_op && dentry->d_op->d_release)
4     dentry->d_op->d_release(dentry);
5   call_rcu(&dentry->d_rcu, d_callback, dentry);
6 }
```

Listing 4. RCU Callback Function for dentries

```
1 static void d_callback(void *arg)
2 {
3   struct dentry * dentry = (struct dentry *)arg;
4
5   if (dname_external(dentry)) {
6     kfree(dentry->d_qstr);
7   }
8   kmem_cache_free(dentry_cache, dentry);
9 }
```

Rename

The d_move() function implements the dentry-specific portion of the rename system call, as shown in Listing 5. Line 5 excludes any other tasks attempting to update dcache, and line 6 permits d_lookup() to determine that it has raced with a rename. Lines 7–13 acquire the per-dentry lock of the file being renamed and its destination, ordered by address so as to avoid deadlock. Lines 14–17 remove the entry from its old location in the dcache hash table, if it has not been so removed already.

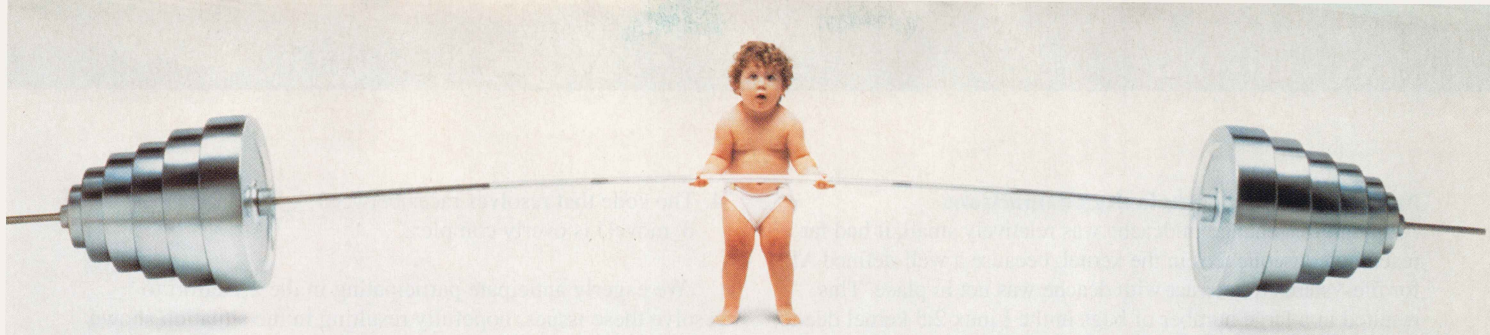
Line 19 updates the dentry to point to its new hash bucket, lines 20–21 add the dentry to its destination hash bucket and line 22 updates the flags to indicate that the dentry is present in the dcache hash table. Line 24 removes the target dentry—the one being rename()ed over—from the dcache hash table, and lines 25–26 divorce the moving and target dentries from their old parents.

Line 27 changes the dentry's name, and line 28 enforces write ordering. The name change is nontrivial due to the fact that short names are stored in the dentry itself, and longer names are stored in separately allocated memory. Lines 29–32 update the name length and hash value. Lines 33–44 connect the dentry to its new parent. Finally, line 45 updates the d_move_count so __d_lookup() can detect races, and lines 46–49 release the locks.

In theory, a sustained succession of rename operations carefully designed to leave dentries in the same directory and in the same hash chain could stall indefinitely horribly unlucky lookups. One way this stall could happen is if the lookup is searching for the last element in the hash chain and the second-to-last element is renamed consistently (thus moved to the head of the list) just as the lookup got to it. In practice, dcache hash chains are short and renames are slow. If these stalls become a problem, though, it may be necessary to add code to stall renames upon path-walk failure. Another approach being considered is to eliminate the global hash table entirely in favor of modifying the d_subdirs list so as to handle large directories gracefully.

Listing 5. Renaming dentries

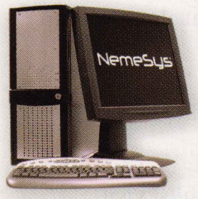
```
1 void
2 d_move(struct dentry *dentry,
3        struct dentry *target)
4 {
5   spin_lock(&dcache_lock);
6   write_seqlock(&rename_lock);
7   if (target < dentry) {
8     spin_lock(&target->d_lock);
9     spin_lock(&dentry->d_lock);
10  } else {
11    spin_lock(&dentry->d_lock);
12    spin_lock(&target->d_lock);
13  }
14  if (dentry->d_vfs_flags & DCACHE_UNHASHED)
15    goto already_unhashed;
16  if (dentry->d_bucket != target->d_bucket) {
17    hlist_del_rcu(&dentry->d_hash);
18 already_unhashed:
19    dentry->d_bucket = target->d_bucket;
20    hlist_add_head_rcu(&dentry->d_hash,
21                       target->d_bucket);
22    dentry->d_vfs_flags &= ~DCACHE_UNHASHED;
23  }
24  __d_drop(target);
25  list_del(&dentry->d_child);
26  list_del(&target->d_child);
27  switch_names(dentry, target);
28  smp_wmb();
29  do_switch(dentry->d_name.len,
30            target->d_name.len);
31  do_switch(dentry->d_name.hash,
32            target->d_name.hash);
33  if (IS_ROOT(dentry)) {
34    dentry->d_parent = target->d_parent;
35    target->d_parent = target;
36    INIT_LIST_HEAD(&target->d_child);
37  } else {
38    do_switch(dentry->d_parent,
39              target->d_parent);
40    list_add(&target->d_child,
41       &target->d_parent->d_subdirs);
42  }
43  list_add(&dentry->d_child,
44     &dentry->d_parent->d_subdirs);
45  dentry->d_move_count++;
46  spin_unlock(&target->d_lock);
47  spin_unlock(&dentry->d_lock);
48  write_sequnlock(&rename_lock);
49  spin_unlock(&dcache_lock);
50 }
```



Experience the most computing power in the smallest footprint possible...

The RackSaver® BladeRack™, powered by Intel® Xeon® processors, is designed to support a wide variety of enterprise solutions...including yours.

Over the past 10 years, RackSaver has provided customers in every industry from Entertainment to Oil and Gas with the most powerful, reliable, and scalable high performance clusters available on the market today. Our customers understand that by increasing density in their server room they improve productivity while lowering their overall total cost of ownership. In addition, RackSaver offers complete onsite enterprise-level support 24 hours a day, 7 days a week to maximize your investment protection. Discover why industry leaders like Industrial Light and Magic, NVIDIA, Shell, WesternGeco, and Novartis rely on RackSaver for all of their high performance computing needs.



THE NEMESYS

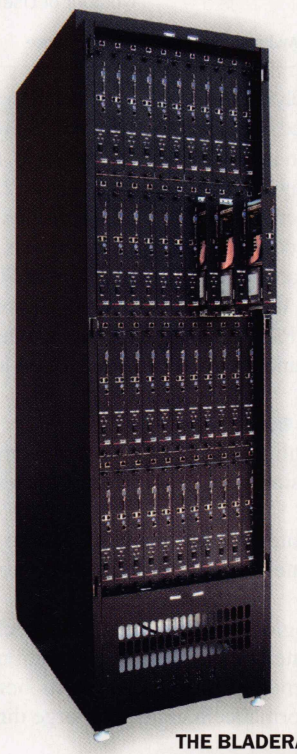
Did we mention that we sell high-end customizable graphics workstations too?

RackSaver has launched a new line of high-end graphic workstations. NemeSys™, powered by the NVIDIA Quadro® FX family, takes the leading design and digital content creation applications to a new level of interactivity. For the first time, styling and production rendering become integral functions of the design workflow, shortening the production process and enabling faster time to market.

THE HIVE CLUSTER

Need a blade solution that can fit under your desk at work and provide more power than ever before?

The HIVE™ houses up to 11 blades that can accommodate a variety of options including high-end servers, SAN, NAS-based storage, along with InfiniBand®. Both the 8 blade and 11 blade versions are fully compatible with RackSaver's RS-1100VM BladeRack line of products. The HIVE's innovative cooling technologies are specifically designed for the needs of small rack systems and also make them suitable for a wide range of environments including military, industrial or a simple office setting.



THE BLADERACK RS-1100VM

RackSaver®
Leader in High-Density Servers!

Start configuring your company's high performance supercomputing solution now by visiting RackSaver online at www.racksaver.com or call one of our Account Managers at (888) 942-3800.

Copyright ©2003, RackSaver Inc. All Rights Reserved. RackSaver, the RackSaver logo, BladeRack, and NemeSys are trademarks or registered trademarks of RackSaver Inc. Intel and Xeon and combinations thereof are trademarks of Intel Corporation or its subsidiaries in the United States or other countries.

Performance and Complexity Comparisons

Although this change in dcache was relatively small, it had far-reaching consequences in the kernel, because a well-defined API for filesystems to interact with dcache was not in place. This resulted in a large number of bugs in the Linux 2.5 kernel due to filesystems hackers attempting to manipulate dcache directly in the traditional style. Given that a somewhat more formal API now exists, we hope future changes will be less traumatic.

Figure 9 shows the performance of a multiuser benchmark running on a Linux 2.5.59 kernel patched to use RCU in the directory-entry cache compared to the performance of an unpatched kernel. These benchmarks were run on a 16-CPU NUMA-Q system using 700MHz PIII Intel Xeons with 1MB L2 cache and 16GB of memory.

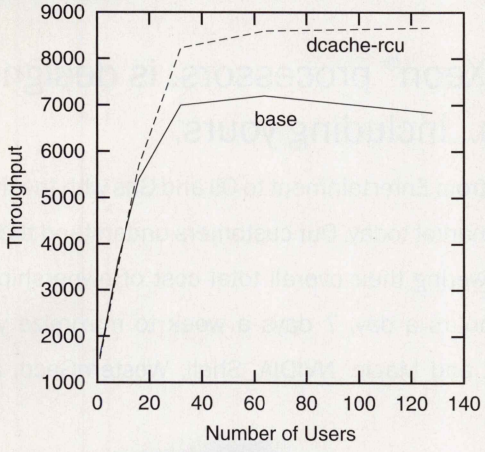


Figure 9. Multiuser Benchmark Performance

Applying the dcache_rcu patch to a Linux 2.4.17 kernel increased SPECweb99 (without SSL) throughput from 2,258 to 2,530 on an 8-CPU PIII Xeon server, a 12% improvement. Applying the same patch to a Linux 2.5.40-mm2 kernel reduced the system time consumed by a Linux kernel build from 47.548 CPU seconds to 42.498 CPU seconds, more than a 10% reduction. A similar test run on a uniprocessor 700MHz PIII Xeon system running the Linux 2.5.42 kernel showed no change. In summary, dcache RCU not only increases scaling for high-end machines, it also maintains good performance on low-end machines.

Future Directions

Although the 2.6 dcache system is much more scalable than the 2.4 version was, a number of issues still need to be investigated:

1. Updates still are gated by dcache_lock, which means that update-intensive workloads do not scale well.

2. The global hash table defeats cache locality and makes update code more complex than necessary. Of course, any alternative must preserve its benefits, including high-performance handling of large directories.

3. The 2.6 dcache code acquires each dentry's d_lock spinlock, resulting in cache-line bouncing and atomic operations, particularly on the root directory and on working directories. Much thought is needed to arrive at a simple solution, as moving permissions into the dentry turns out to be quite complex.

4. The code that resolves races between _ _d_lookup() and d_move() is overly complex.

We eagerly anticipate participating in the 2.7 effort to resolve these issues, hopefully resulting in the situation shown in Figure 10.

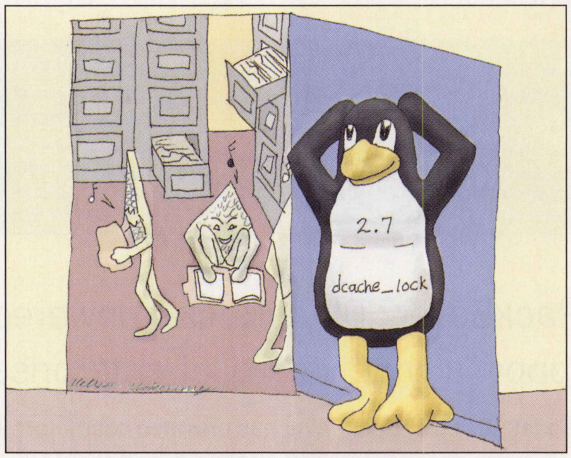


Figure 10. Tux's Duty in 2.7

Legal Statement

This work represents the view of the author and does not necessarily represent the view of IBM.

SPEC and the benchmark name SPECweb are registered trademarks of the Standard Performance Evaluation Corporation. The benchmarking was done for research purposes only and may not be compared to published results on the SPECWeb site, due to the following deviations from the rules:

1. It was run on hardware that does not meet the SPEC availability-to-the-public criteria. The machine was an engineering sample.

2. access_log was not kept for full accounting. It was being written but deleted every 200 seconds.

For the latest SPECweb99 benchmark results, visit www.spec.org.

Paul E. McKenney is a distinguished engineer at IBM and has worked on SMP and NUMA algorithms for longer than he cares to admit. Prior to that, he worked on packet-radio and Internet protocols (but long before the Internet became popular). His hobbies include running and the usual house-wife-and-kids habit.

Dipankar Sarma currently is working on a number of Linux kernel projects, including CPU hot-plug, RCU and VFS enhancements. Prior to his Linux days, he worked on a number of areas including ABI, OS bringup, I/O drivers and multipath I/O.

Maneesh Soni has been working with IBM's Linux Technology Center as a member of Linux Scalability Effort Project. He has experience in the system software arena, particularly with operating-system kernels and filesystems.

The only thing more impressive than the price of our Dedicated Hosting solutions is the performance.

High Performance Plans 100 – 400	
Hardware:	IBM eserver xSeries
CPU:	2.4 GHz Pentium® 4 – Dual 2.67 GHz Xeon
RAM:	512 MB – 2 GB
Storage:	60 GB – Three 36.4 GB SCSI
Monthly Transfer: (with Burstable Bandwidth)	100 GB – 200 GB
Operating Systems:	Red Hat Linux 9.0 or Windows Server 2003
Pricing:	Plans starting at $349/month with no setup fee

The way Web hosting should be.

The performance you need, support you can count on: Interland has it all. With our range of High Performance solutions, you can run the latest, most reliable applications. And now all this performance is packaged in a great deal.

SAVE UP TO $1,049

Get FREE Setup and Price Match Guarantee when you buy any High Performance solution

Call 1-800-844-9938 or go to www.interland.com/dedicated

Of course, everything's backed by Interland's 24/7 High Value Support team. We have certified hosting veterans in Microsoft, Red Hat, IBM, Cisco – virtually any operating system or hardware. And you can take advantage of innovative programs like ValueGuard – standard with all High Performance plans – which updates your server solution to the latest High Performance configurations in 18 months. Also standard is secure access to mymanagedhosting.com, our online support portal where you can view statements, submit and track trouble tickets, receive network and security updates, and much more. No wonder we're the world leader in Web hosting for small and medium businesses.

Interland®

INTERLAND PRICE MATCH GUARANTEE

IBM Business Partner

Microsoft CERTIFIED Partner

redhat. CERTIFIED ENGINEER

© 2003 Interland, Inc. All Rights reserved. Not responsible for errors or omissions in typography or photography. Interland is a registered trademark and the Interland logo is a trademark of Interland, Inc. All orders subject to credit approval, availability, and other terms and conditions (including terms of service and acceptable use policy) available at www.interland.com. Offer valid for a limited time only. With the purchase of a new High Performance dedicated account, customer will receive a waiver of the standard setup fees for such account. Savings of up to $1,049 reflect the waived setup fee for the Windows High Performance 400 plan. Promotion valid only for telephone sales and cannot be combined with any other offers.

Signed Kernel Modules

Now you can make the kernel check modules for a cryptographic signature before inserting them. Here's the detail of how it's done.

BY GREG KROAH-HARTMAN

Signed kernel modules have been a feature of other operating systems for a number of years. Some people and companies like the idea of installing only modules (or drivers, as they are sometimes called) that are known to be blessed by some authority in their operating systems. Given the changes in how Linux loads kernel modules, signed kernel modules easily can be added to the Linux kernel. This article discusses how I have implemented this feature and details how to use it.

In a signed kernel module, someone has inserted a digital signature into the module stating they trust this specific module. I am not going to try to persuade anyone that Linux should have this ability, that it should be required or even that it provides increased security. I describe only how to do it and provide the method for its implementation, if anyone wants to use it.

Public key cryptography is used to make signed kernel modules work. For an overview of the RSA public-key cryptographic algorithm—what it is and how it works—see the *Linux Journal* Web article at www.linuxjournal.com/article/6826. This article assumes readers are familiar with the basics of public key cryptography and that they are able to patch, build and load a new Linux kernel onto their machines. For instructions on how to build and load a new kernel, see the very helpful Linux Kernel HOWTO located at www.tldp.org.

In the 2.5 kernel development series, Rusty Russell rewrote the way Linux kernel modules work. In previous kernels, the majority of the module loading logic was stored in user space. With Rusty's changes, all of that logic moved into the kernel, reducing the amount of architecture-independent logic and simplifying the user interface greatly. One nice side benefit of this is the kernel now has access to the entire module file in raw form. The kernel module simply is a file in ELF format. ELF stands for executable and linking format and is the format used for executable programs. The ELF specification can be found in text form at www.muppetlabs.com/~breadbox/software/ELF.txt.

ELF files are comprised of different sections. These sections can be seen by running the readelf program. For example:

```
$ readelf -S visor.ko
There are 23 section headers, starting at offset 0x3954:
Section Headers:
  [Nr] Name              Type            Addr     Off    Size   ES Flg Lk Inf Al
  [ 0]                   NULL            00000000 000000 000000 00      0   0  0
  [ 1] .text             PROGBITS        00000000 000040 0017e0 00  AX  0   0 16
  [ 2] .rel.text         REL             00000000 003cec 000cd0 08      21  1  4
  [ 3] .init.text        PROGBITS        00000000 001820 000210 00  AX  0   0 16
  [ 4] .rel.init.text    REL             00000000 0049bc 0001c8 08      21  3  4
  [ 5] .exit.text        PROGBITS        00000000 001a30 000030 00  AX  0   0 16
  [ 6] .rel.exit.text    REL             00000000 004b84 000030 08      21  5  4
  [ 7] .rodata           PROGBITS        00000000 001a60 000020 00   A  0   0 16
  [ 8] .rel.rodata       REL             00000000 004bb4 000028 08      21  7  4
  [ 9] .rodata.str1.1    PROGBITS        00000000 001a80 000449 01 AMS  0   0  1
  [10] .rodata.str1.32   PROGBITS        00000000 001ee0 0009c0 01 AMS  0   0 32
  [11] .modinfo          PROGBITS        00000000 0028a0 0006c0 00   A  0   0 32
  [12] .data             PROGBITS        00000000 002f60 000600 00  WA  0   0 32
  [13] .rel.data         REL             00000000 004bdc 0001e0 08      21  c  4
  [14] .gnu.linkonce.thi PROGBITS        00000000 003560 000120 00  WA  0   0 32
  [15] .rel.gnu.linkonce REL             00000000 004dbc 000010 08      21  e  4
  [16] __obsparm         PROGBITS        00000000 003680 000180 00  WA  0   0 32
  [17] .bss              NOBITS          00000000 003800 00000c 00  WA  0   0  4
  [18] .comment          PROGBITS        00000000 003800 00006e 00      0   0  1
  [19] .note             NOTE            00000000 00386e 000028 00      0   0  1
  [20] .shstrtab         STRTAB          00000000 003896 0000bd 00      0   0  1
  [21] .symtab           SYMTAB          00000000 004dcc 000760 10      22 58  4
  [22] .strtab           STRTAB          00000000 00552c 000580 00      0   0  1
```

Because ELF files are made up of sections, it is easy to add a new section to the module file and have the kernel read it into memory when it tries to load the module. If we put an RSA-signed section into the module, the kernel can decrypt the signature and compare it to the signature of the file it just loaded. If it matches, the signature is valid and the module is inserted successfully into the kernel's memory. If the signature does not match, either something has been tampered with in the module or the module was not signed with a proper key. The module then can be rejected—that is what my patch does.

How the Kernel Code Works

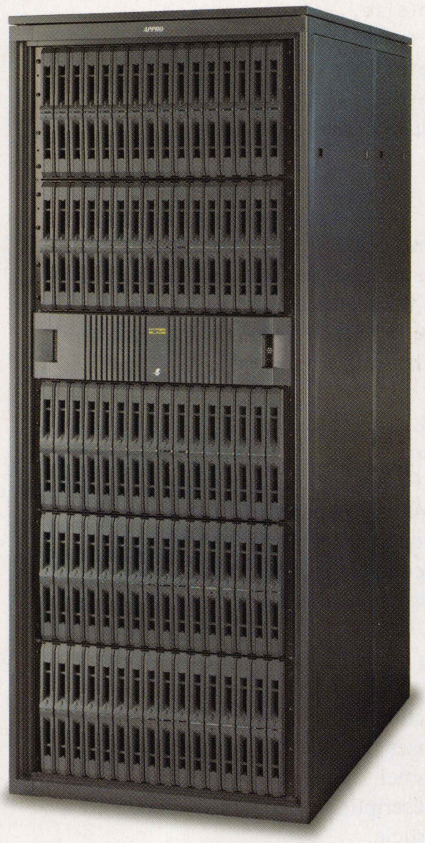
When the kernel is told to load a module, the code in the file kernel/module.c is run. In that file, the function load_module does all of the work of breaking the module into the proper sections, checking memory locations, checking symbols and all

Appro High-Performance Servers & Cluster Solutions

Appro's servers based on AMD Opteron™ processors boost performance on existing Windows® & Linux applications.

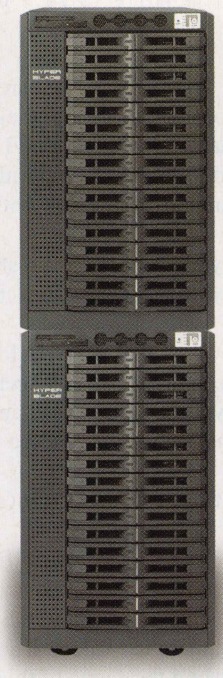
Double the Density!

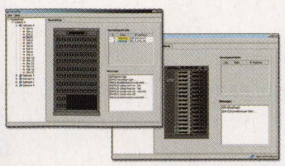
Appro *Hyper*Blade Fully Integrated Cluster Solutions



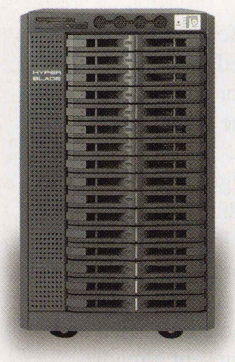
New!

Mini-Cluster



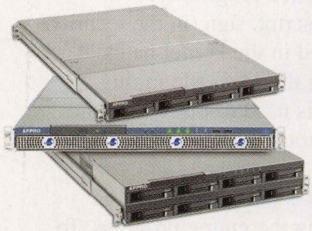


Appro BladeDome Remote Server Management Option



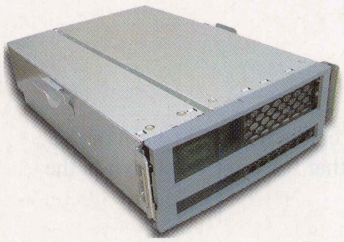
- Flexible, modular and scalable architecture for rapidly changing environment
- Blade servers fit Appro full-size cabinet (up to 80-nodes) and Appro mini-size cabinet (up to 17-nodes) with the ability to mix and match cluster building blocks
- Dual AMD Opteron™ processors per node
- Up to 16GB ECC DDR Memory per node
- One PCI-X slot per node
- Up to two internal HDDs
- Remote Management Options:
  - In-band and out-of-band control
  - GUI & Command Line Interfaces
- High-Speed Interconnect Options: Myrinet, Dolphin, Quadrics and Infiniband

Appro 1U and 2U Servers



- Dual AMD Opteron™ processors
- Design to run cool and efficiently in demanding rack conditions
- High quality, powerful and cost-effective
- Fully integrated cluster solutions available

Appro 4U - 4 Way Server



- Quad AMD Opteron™ processors
- Memory capacity and flexibility to support a wide range of applications
- Supports up to 64GB in 16 DIMM sockets
- Supports up to 4 SCSI drives
- 3.5" floppy drive and slim CD-ROM

AMD Opteron™ Processors - Shorten run-time cycles and increase bandwidth for processing computing requests.
- 32 bit applications while you migrate to 64 bit computing for long-term investment protection.

APPRO
HPC Cluster Solutions

For more information, please visit **www.appro.com**
or call Appro Sales at **800.927.5464, 408.941.8100.**

AMD 64 Opteron

the other tasks a linker generally does. The patch modifies this function and adds the following lines of code:

```
if (module_check_sig(hdr, sechdrs, secstrings)) {
    err = -EPERM;
     goto free_hdr;
}
```

This new function, module_check_sig, does all of the module signature-checking logic. If it returns an error, the error `Improper Permission` is returned to the user and module loading is aborted. If the function returns a 0, meaning no error occurred, the module load procedure continues on successfully.

The module_check_sig function is located in the file kernel/module-sig.c. The first thing the function does is check to see if a signature is located within the module. This is done with the following lines of code:

```
sig_index = 0;
for (i = 1; i < hdr->e_shnum; i++)
     if (strcmp(secstrings+sechdrs[i].sh_name,
                "module_sig") == 0) {
          sig_index = i;
          break;
}
if (sig_index <= 0)
     return -EPERM;
```

This bit of code loops through all of the different ELF sections in the kernel module and looks for one called module_sig. If it does not find the signature, it returns an error and prevents this module from being loaded. If it does find the signature, the function continues.

Once the kernel has found the module signature, it needs to determine what the hash value is of the module it is being asked to load. To do this, it generates the SHA1 hash of the ELF section that contains executable code or data used by the kernel. The kernel already contains code to generate SHA1 hashes (along with other kinds of hashes, including MD5 and MD4), so most of the logic for this step is present already.

The function first allocates a crypto transformation structure by requesting the SHA1 algorithm. It then initializes this structure with the following lines of code:

```
sha1_tfm = crypto_alloc_tfm("sha1", 0);
if (sha1_tfm == NULL)
     return -ENOMEM;
crypto_digest_init(sha1_tfm);
```

The sha1_tfm variable is used to create the SHA1 hash of the specific portions of the ELF file that we want, as shown in the following code:

```
for (i = 1; i < hdr->e_shnum; i++) {
     name = secstrings+sechdrs[i].sh_name;

     /* We only care about sections with "text" or
        "data" in their names */
     if ((strstr(name, "text") == NULL) &&
          (strstr(name, "data") == NULL))
          continue;
     /* avoid the ".rel.*" sections too. */
     if (strstr(name, ".rel.") != NULL)
          continue;

     temp = (void *)sechdrs[i].sh_addr;
     size = sechdrs[i].sh_size;
     do {
          memset(&sg, 0x00, sizeof(*sg));
          sg.page = virt_to_page(temp);
          sg.offset = offset_in_page(temp);
          sg.length = min(size,
                          (PAGE_SIZE - sg.offset));
          size -= sg.length;
          temp += sg.length;
          crypto_digest_update(sha1_tfm, &sg, 1);
     } while (size > 0);
}
```

In this code, we care only about the ELF sections with the word text or data in their names but not ones that contain the characters .rel. After all of the sections have been found and fed to the SHA1 algorithm, the SHA1 hash is placed into the variable sha1_result with the following lines:

```
crypto_digest_final(sha1_tfm, sha1_result);
crypto_free_tfm(sha1_tfm);
```

Now that the SHA1 hash is computed and the place with the signed hash has been found, all that is left to do is unencrypt the signed hash and compare it to the calculated one. This step is done in the last line of this function:

```
return rsa_check_sig(sig, &sha1_result[0]);
```

The rsa_check_sig function is located in the security/rsa/rsa.c file and uses the GnuPG code itself, which was ported to run in the kernel to unencrypt the signature and compare the values. The description of how this works is beyond the scope of this article.

How the User-Space Code Works

Now that we have seen how the kernel determines whether a module is signed properly, how do we get a signature into a module in the first place? Two user-space programs, extract_pkey and mod, and one small script, sign (in the security/rsa/userspace/ directory), can be found in the kernel patch. The two programs can be built by running the Makefile in this directory. The extract_pkey program is used to place a public key into the kernel, and the mod program is used by the sign script to sign a kernel module.

In order to sign a module, an RSA-signing key must be generated, which can be done by using the gnupg program. To generate an RSA-signing key, pass the --gen-key option to gpg:

```
$ gpg --gen-key
gpg (GnuPG) 1.2.1; Copyright (C) 2002 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.
```

```
Please select what kind of key you want:
   (1) DSA and ElGamal (default)
   (2) DSA (sign only)
   (5) RSA (sign only)
Your selection?
```

We want to create an RSA key, so we select option 5 and then choose the default key size of 1024:

```
Your selection? 5
What keysize do you want? (1024)
Requested keysize is 1024 bits
```

Continue answering the rest of the questions, and eventually your RSA key is generated. But in order to use this key, we must create an encrypting version of it. To do that, run gpg again and edit the key you just created (in the text below, I have named my key testkey):

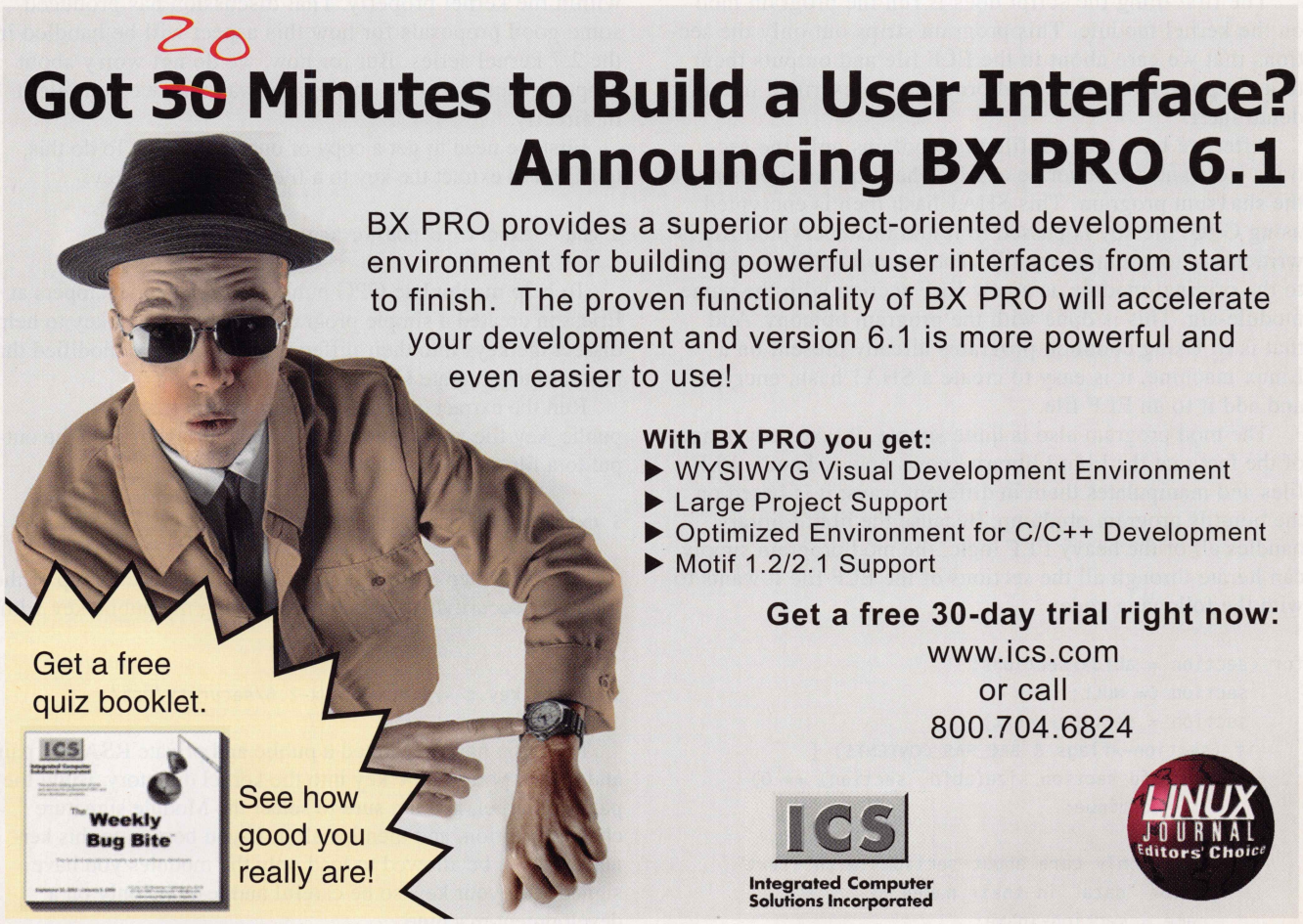
```
$ gpg --edit-key testkey
gpg (GnuPG) 1.2.1; Copyright (C) 2002 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

Secret key is available.

gpg: checking the trustdb
gpg: checking at depth 0 signed=0 ot(-/q/n/m/f/u)=0/0/0/0/0/1
pub  1024R/77540AE9  created: 2003-10-09 expires: never trust: u/u
(1). testkey

Command>
```

We want to add a new key, so type addkey at the prompt:

```
Command> addkey
Please select what kind of key you want:
   (2) DSA (sign only)
   (3) ElGamal (encrypt only)
   (5) RSA (sign only)
   (6) RSA (encrypt only)
Your selection?
```

Again, we want an RSA key, so choose option 6 and answer the rest of the questions. After the key is generated, type `quit` at the prompt:

```
Command> quit
Save changes? yes
```

Copyright 2002 Integrated Computer Solutions, Inc. All rights reserved, all trademarks are the property of their respective companies.

Now that we have a key, we can use it to sign a kernel module.

To sign a module, use the sign script, which is a simple shell script:

```
#!/bin/bash
module=$1
key=$2

# strip out only the sections that we care about
./mod $module $module.out

# sha1 the sections
sha1sum $module.out | awk "{print \$1}" > \
$module.sha1

# encrypt the sections
gpg --no-greeting -e -o - -r $key $module.sha1 > \
$module.crypt

# add the encrypted data to the module
objcopy --add-section module_sig=$module.crypt \
$module

# remove the temporary files
rm $module.out $module.sha1 $module.crypt
```

The first thing the script does is run the program mod on the kernel module. This program strips out only the sections that we care about in the ELF file and outputs them to a temporary file. The mod program is described in more detail later.

After we have an ELF file that contains only the sections we want, we generate a SHA1 hash of the file using the sha1sum program. This SHA1 hash then is encrypted using GPG, the key is passed to it and this encrypted file is written out to a temporary file. The encrypted file is added to the original module as a new ELF section with the name module-sig. This is done with the program objcopy. And that is it. Using common programs already present on a Linux machine, it is easy to create a SHA1 hash, encrypt it and add it to an ELF file.

The mod program also is quite simple. It takes advantage of the fact that the libbfd library knows how to handle ELF files and manipulates them in different ways; it is based on the binutils program objdump. Because the libbfd library handles all of the heavy ELF logic, the mod program simply can iterate through all the sections of the ELF file it wants to with the following code:

```
for (section = abfd->sections;
     section != NULL;
     section = section->next) {
    if (section->flags & SEC_HAS_CONTENTS) {
        if (bfd_section_size(abfd, section) == 0)
            continue;

        /* We only care about sections with "text"
           or "data" in their names */
        name = section->name;
        if ((strstr(name, "text") == NULL) &&
            (strstr(name, "data") == NULL))
            continue;

        size = bfd_section_size(abfd, section));
        data = (bfd_byte *)malloc(size);

        bfd_get_section_contents(abfd, section,
                                 (PTR)data,
                                 0, size);

        stop_offset = size / opb;

        for (addr_offset = 0;
             addr_offset < stop_offset;
             ++addr_offset) {
            fprintf(out, "%c", data[addr_offset]);
        }
        free(data);
    }
}
```

Now that we can sign a kernel module and the kernel knows how to detect this signature, the only remaining piece is to put our public key into the kernel so it can decrypt the signature successfully. A lot of discussion on the linux-kernel mailing list recently has centered on how to handle keys within the kernel properly. That discussion has produced some good proposals for how this aspect will be handled in the 2.7 kernel series. But for now, we do not worry about properly handling keys in flexible ways, so we compile it in directly.

First we need to get a copy of our public key. To do this, tell GPG to extract the key to a file called public_key:

```
$ gpg --export -o public_key
```

To help manipulate GPG public keys, some developers at Ericsson created a simple program called extract_pkey to help dissect the keys into their different pieces. I have modified that program to generate C code for the public key.

Run the extract_pkey program and point it at the public_key file you generated previously. Have it send the output to a file called rsa_key.c:

```
$ extract_pkey public_key > rsa_key.c
```

After this step is finished, move that rsa_key.c on top of the file in the security/rsa/ directory, replacing my public key with yours:

```
$ mv rsa_key.c ~/linux/linux-2.6/security/rsa/
```

Now you have generated a public and private RSA key pair and placed your public key into the kernel directory. Build the patched kernel, making sure to select the Module signature checking option, and then install it. If you boot in to this kernel, you will be allowed to load only the modules you have signed with your key, so be careful and test this only on a development machine.

What Is Left to Do?

As shown in this article, a number of different steps are required to generate a key, sign a kernel module and place the public key into the kernel image. This still is a rough development project. In order to make it more acceptable to the kernel developers and to the Linux community in general, these steps need to be automated, making it easier to sign all kernel modules and handle the public key.

Besides the obvious need to simplify the use of this feature, some other future goals of this project include:

- Move the RSA code into the generic crypto framework, allowing other kernel features to use it.
- Allow more than one public key to be present in the kernel, letting multiple sources of signed kernel modules run in a single machine.
- Simplify the signing logic to allow GPG's native signing functionality or possibly the functionality provided in the bsign program to be used, instead of the custom mod program.

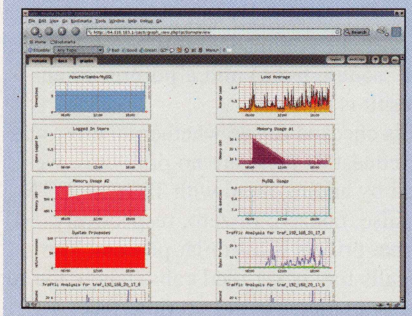
Acknowledgements

I would like to thank the developers at Ericsson, who have created a kernel patch and program called digsig, for allowing me to use their port of GPG to the kernel. I previously had done this, but the implementation was horrible; thankfully, they released their port and were very helpful. The digsig kernel patch allows users to sign programs and prevents the kernel from running any program not signed. More information about this project can be found at sourceforge.net/projects/disec.

I also would like to thank my employer, IBM, for allowing me to work on this project, and Don Marti, for prodding me to finish it and write this article.

Greg Kroah-Hartman currently is the Linux kernel maintainer for a variety of different driver subsystems. He works for IBM, doing Linux kernel-related things, and can be reached at greg@kroah.com.

Cacti: www.raxnet.net/products/cacti



Cacti is similar in some ways to MRTG but is easier to set up and administer and, by default, graphs more than traffic. But, some Cacti users haven't stopped there; they've graphed Web server hits, bind statistics, weather and more. A quick look at the Cacti home page is a real eye-opener. I'm sure you could use it to graph even more with a few quick scripts. Requires: rrdtool, MySQL, Web server, PHP with SNMP and MySQL support, snmpd, Perl and a graphical Web browser.

—DAVID A. BANDEL

www.asacomputers.com

Hardware Systems for the Open Source Community—since 1989

(Linux, FreeBSD, NetBSD, OpenBSD, Solaris...)

4TB in 4U—$8,299
Intel E7501, 2 x Intel Xeon 2.4GHz
1 GB ECC DDR *Max: 16GB*
1 x 10/100 LAN
1 Gigabit LAN
Raid 0,1,5,etc.
CD, Floppy
Redundant Hot-Swap Power Supply

2U Dual Xeon SCSI—$2,260
6 Front-removable Hot-Swap SCSI bays
2 x Intel Xeon 2.4 GHz *Max: 2*
1 GB ECC DDR *Max: 16GB*
1 x 36GB U320 SCSI Drives
Raid optional *Max: 876GB Storage*
CD, Floppy
Dual Gig ports
1 Extra 5.25 Drive Bay
Hot-plug redundant power supply

Dual AMD Opteron in 1U—$1,729
2 x AMD Opteron 240
1 GB DDR ECC Registered Memory
1 x 40 GB IDE
Gigabit LAN

AMD Opteron™

Space-saving Mini-1U Dual Xeon IDE—$1,529
1U 14" Depth
533 MHz FSB
2 x Intel Xeon 2.4 GHz
1 GB DDR ECC Memory
40 GB IDE HDD
1 GIG, 1 10/100 LAN

1U P4 Entry-level Server—$735
1 x Intel P4 2.4 GHz
256 MB DDR ECC Memory
40 GB IDE Drive
CD, 2 x 10/100 LAN

1U Dual Xeon SCSI—$1,558
SuperMicro 6013P-8 Barebones
Dual Intel Xeon 2.4 GHz 533 FSB
512 MB DDR ECC *Max: 12GB*
36 GB 10K rpm SCSI
3 Hot-Swap SCSI Drive Bays
CD, Floppy
Dual Gig, VGA onboard
Raid optional

ASA Colocation
$50 per month for
1U Rack - 20 GB/month

ASA Colocation Special
First month of colocation free.*

Storage Solutions
IDE, SCSI, Fiber RAID solutions
TB storage options
3Ware, Promise, Adaptec,
ICP-Vortex, LSI, CMD, Chaparral,
Mylex controllers
JMR, Kingston/Storcase solutions

Clusters
Rackmount and Desktop nodes
HP, Intel, 3Com, Cisco switches
KVM or Cyclades Terminal Server
APC or Generic racks

All systems installed and tested with user's choice of Linux distribution (free). ASA Colocation—$50 per month

ASA Computers, Inc.
1-800-REAL-PCS
2354 Calle Del Mundo, Santa Clara, CA 95054
TEL: (408) 654-2900 • FAX: (408) 654-2910
E-MAIL: sales@asacomputers.com
*Free delivery in Silicon Valley**
Open on Sat. too—10 to 1
Prices and availability of systems subject to change without notice
** Certain restrictions apply.*

www.asacomputers.com
www.b2bforsale.com

Controlling Hardware with ioctls

Once you learn the ioctl system call, you'll be able to check the status of the Ethernet link light and other miscellaneous but important facts about hardware. **BY LISA CORSETTI**

A few years ago, I had a laptop that I used at work and at home. To simplify my network configuration and not have to change it manually depending on where I was, I decided to use DHCP in both places. It was the standard at work, so I implemented a DHCP server at home. This worked well except when I booted the system without it being plugged in to either network. When I did, the laptop spent a lot of time trying to find a DHCP server without success before continuing the rest of the startup process.

I concluded that an ideal solution to this lag time would be for the system to start with the Ethernet interface down and have it come up if, and only if, the cable was connected to a hub, that is, if I had a link light on the Ethernet interface. The best way to do this appeared to be having a shell script call a program whose return code would indicate whether a link had been established on a particular network interface. So began my quest for a method to determine this link status of my 10/100Base-T interface.

Not having done much low-level Linux programming, it took me a bit of time to discover that most of this type of interaction with device drivers usually is done through the ioctl library call (an abbreviation of I/O control), prototyped in sys/ioctl.h:

```
int ioctl(int, int, ...)
```

The first argument is a file descriptor. Because all devices in Linux are accessed like files, the file descriptor used usually is one that has been opened with the device to which you are interfacing as the target. In the case of Ethernet interfaces, however, the fd simply is an open socket. Apparently, no need exists to bind this socket to the interface in question.

The second argument in ioclt.h is an integer that represents an identification number of the specific request to ioctl. The requests inherently must vary from device to device. You can, for example, set the speed of a serial device but not a printer device. Of course, a specific set of commands exists for network interfaces.

Additional arguments are optional and could vary from the ioctl implementation on one device to the implementation on another. As far as I can tell, a third argument always is present, and I have yet to find more than a third. This third argument usually seems to be a pointer to a structure. This allows the passing of an arbitrary amount of data in both directions, the data being defined by the structure to which the pointer refers, simply by passing the pointer.

A basic example of how ioctl works is shown in the following simple program that checks the status of one signal on a serial port:

```
#include <termios.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/ioctl.h>

main()
{
  int fd, status;

    fd = open("/dev/ttyS0", O_RDONLY);
    if (ioctl(fd, TIOCMGET, &status) == -1)
       printf("TIOCMGET failed: %s\n",
              strerror(errno));
    else {
       if (status & TIOCM_DTR)
          puts("TIOCM_DTR is not set");
       else
          puts("TIOCM_DTR is set");
    }
    close(fd);
}
```

This program opens a tty (serial port) and then calls ioctl with the fd of the serial port, the command **TIOCMGET** (listed as `get the status of modem bits`) and a pointer to an integer into which the result is returned.

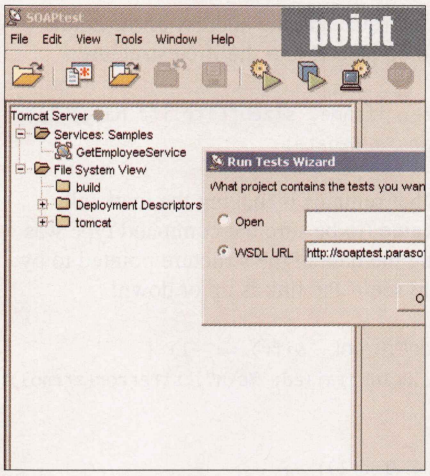
The ioctl result then is checked to see whether an error was made in processing the request. If there are no problems, we check the values returned by anding them with TIOCM_DTR. This step yields true or false, nonzero or zero, respectively.

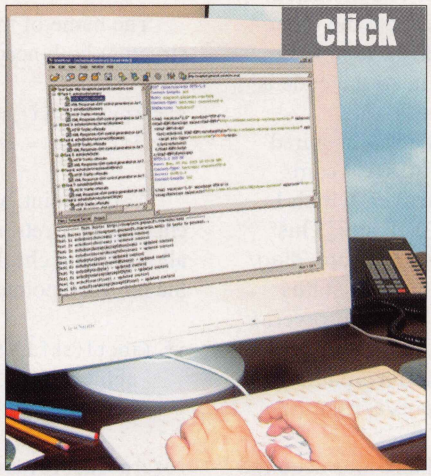
Using ioctl for Ethernet drivers is a similar process. The third parameter to ioctl calls for socket ioctl calls (where the fd is a socket handle) often is a pointer to a ifreq (interface request) structure. The type deceleration for ifreq structures can be found in net/if.h.

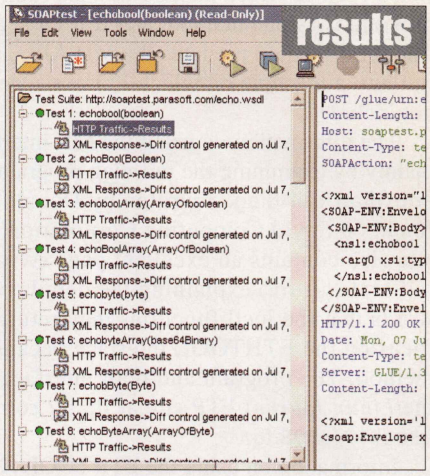
Unfortunately, documentation for many of the ioctl interfaces is difficult to find, and there are at least three different APIs for accessing network interfaces. I originally wrote this program using the MII (media independent interface) method.

PARASOFT
We make software work.™

Verify Web services instantly with Parasoft SOAPtest.







(It's as easy as 1-2-3.)

Parasoft SOAPtest is the first and only testing tool that enables you to instantly verify that your Web services are working properly.

Just point SOAPtest to your Web services WSDL files. It automatically creates test suites and runs comprehensive tests of the Web service's functionality and load capacities.

Server emulation for SOAP client development.
If you're developing a client-side application and need to verify functionalities such as credit card processing, oftentimes the Web services server is in use or otherwise unavailable for testing. Parasoft SOAPtest solves this problem by enabling you to emulate server responses and verify client functionality in just seconds.

SOAPtest also lets you quickly create templates written in Java,™ JavaScript, and Python.

For Downloads go to www.parasoft.com/lj_11. Or call 888-305-0041.

Copyright ©2003 Parasoft Corporation. All rights reserved. All Parasoft product names are trademarks or registered trademarks of Parasoft Corporation in the United States and other countries. All other marks are the property of their respective owners.

Now supports WS-I Basic Profile 1.0

Compatibility:
Works with any SOAP-compliant implementation, including Microsoft and IBM
Windows 2000/XP
Linux
Solaris

A part of Parasoft Automated Error Prevention (AEP) Solutions and Services

PARASOFT
soaptest®

While writing this article, with the most recent kernel installed on my machine, however, I discovered I had to add the ETHTOOL method.

After adding ETHTOOL, I then modified the program and wrote each interface method as a subroutine. The modified program tries one method, and if it fails, attempts the other. The third method predates the MII API, and I have not yet run into a machine on which I have needed it, so the code is not included.

The information on using the MII interface was acquired mainly by examining the mii-diag program (ftp.scyld.com/pub/diag/mii-diag.c) written by Donald Becker, which I found on Scyld Computing Corporation's Web site. This site also contains an excellent page (www.scyld.com/diag/mii-status.html) explaining the details of the MII status words that the ioctl functions may return. Here, however, I focus on the ETHTOOL interface because it is the newer method. The program and both interfaces are available from the *Linux Journal* FTP site at ftp.ssc.com/pub/lj/listings/issue117/6908.tgz.

Information on using the ETHTOOL API also was acquired by scouring various pieces of source code, not the least of which was the source code for the network interface drivers themselves—particularly eepro100.c. Also helpful was an e-mail written by Tim Hockin that I found while Googling.

In writing my program, I set the default interface to eth0 unless a parameter was passed to the program. The interface ID is stored in ifname. Because the ioctl commands I use are specific to network interfaces, using some other device likely will cause a "cannot determine status" to be returned.

Before calling ioctl we need a file handle, so we first must open a socket:

```
int skfd;
if (( skfd = socket( AF_INET, SOCK_DGRAM, 0 ) ) < 0 )
   {
   printf("socket error\n");
   exit(-1);
   }
```

In the standard try-to-check-for-all-errors C coding style, I placed this inside an if statement that simply prints an error and terminates the program, returning a -1 if the socket does not open properly. For my purposes, I would rather report errors in determining status as a lack of a link rather than as a presence of one, so a found link is reported as 0 and not found is reported as 1.

The new ETHTOOL API for interfacing to the driver has made determining the status of the link much easier than did the previous method. ioctl was implemented for ETHTOOL interfaces such that there is now only ONE ioctl command, SIOCETHTOOL (which specifies that the call is an ETHTOOL command), and the data block passed then contains the specific subcommand for the ETHTOOL interface.

The standard ioctl data structure (type ifreq) requires two additional items: the name of the interface to which the command should be applied and an address of a structure (type ethtool_value) in which to store the specific command as well as the returned information.

The structures and most other information (including the commands available) are located in the ethtool.h header file. The command that I needed was ETHTOOL_GLINK, documented as "get link status", which I stored in edata.cmd:

```
edata.cmd = ETHTOOL_GLINK;
```

The name of the interface and the address of the edata structure both need to be placed into the ifreq structure:

```
strncpy(ifr.ifr_name, ifname, sizeof(ifr.ifr_name)-1);
ifr.ifr_data = (char *) &edata;
```

At this point, all that remains is making the ioctl call, checking the return value (to be sure the command type was allowed) and checking the data in the structure pointed to by the returned pointer to see if the link is up or down:

```
if (ioctl(skfd, SIOCETHTOOL, &ifr) == -1) {
    printf("ETHTOOL_GLINK failed: %s\n", strerror(errno));
    return 2;
}
return (edata.data ? 0 : 1);
```

In this case, my code returns a 0 for link up, a 1 for link down and a 2 for undetermined or failure. This code allows me to call this function from my rc.local, bring the interface up and either call dhcpcd or pump to get an IP address only if the system is plugged in to a functioning hub/switch. Here is the relevant section of rc.local:

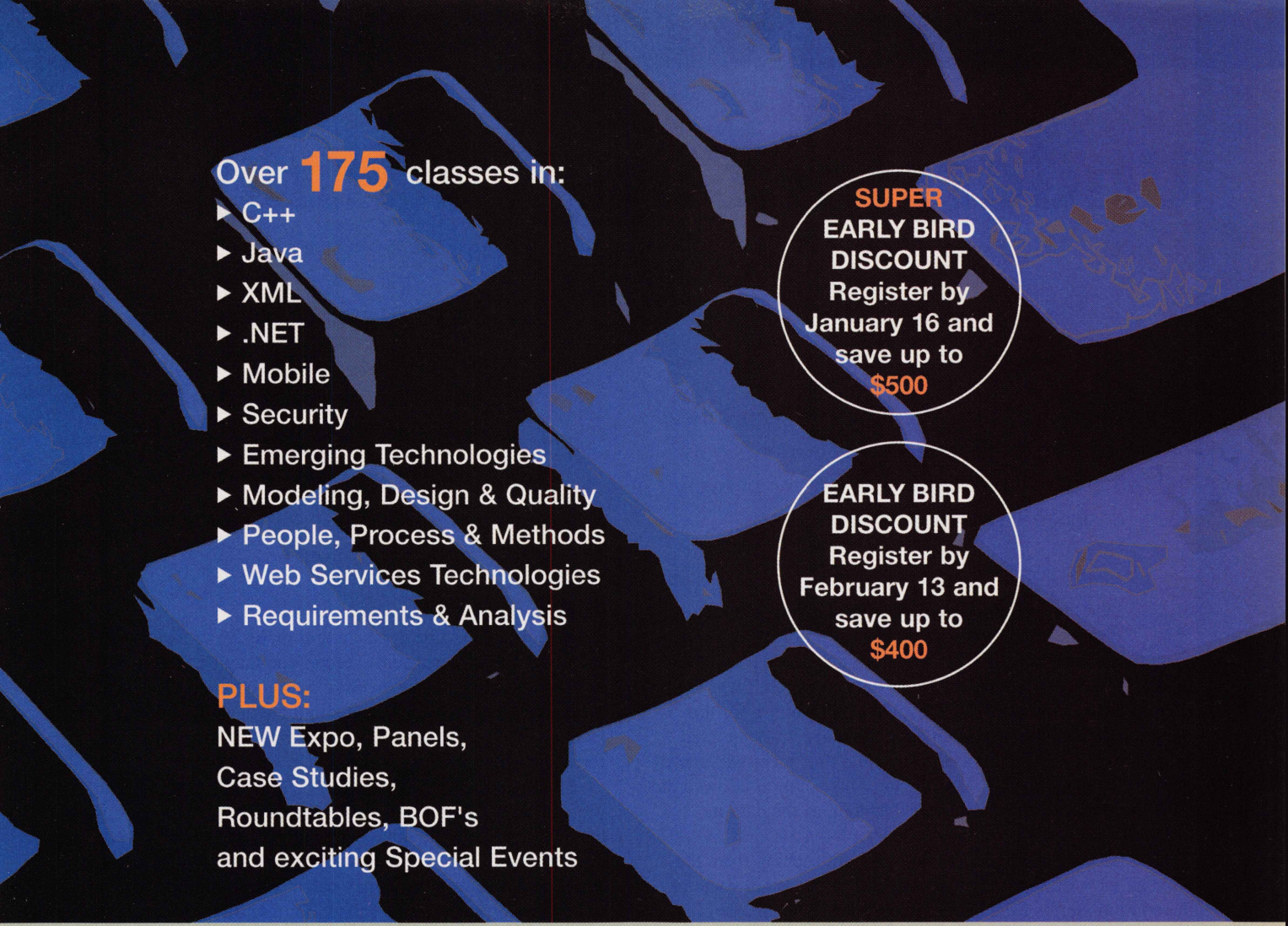
```
/root/sense_link/sense_link | logger
if /root/sense_link/sense_link > /dev/null; then
  logger "No link sense -- downing eth0"
  /sbin/ifconfig eth0 down
else
  logger "Sensed link - dhcping eth0"
  /sbin/dhcpcd eth0
fi
```

First, the output of sense_link is sent to the system log. Then, if no link was sensed on eth0, or if it could not be determined, a message is written to the log and eth0 is taken down. If a link was sensed, dhcpcd is executed on eth0.

Once this is implemented, my rc.local file now executes quite quickly when no network cable is plugged in or when a DHCP server is active and found. The only time I still experience significant delays is if I am plugged in to a network where there is no active DHCP server.

I haven't yet tried this code with my 802.11b card to see if it can detect a link on it before attempting to contact a DHCP server, because I usually only have the PCMCIA card plugged in when I am in a location that I know has a server. It would be an interesting experiment and a useful extension, however, for an interested party.

Lisa Corsetti presently is a software architect as well as the president of Anteil, Inc., a consulting firm that focuses on custom Web-based applications for various industries and government. Lisa received a BS in Electrical and Computer Engineering from Drexel University.



SD WEST 2004 SOFTWARE DEVELOPMENT CONFERENCE & EXPO

MARCH 15-19, 2004, SANTA CLARA, CA

Understanding Caching

Architectures that support Linux differ in how they handle caching at the hardware level. Here's how the kernel gets the best possible use out of every cache design. **BY JAMES BOTTOMLEY**

Since the earliest days of microprocessors, system designers have been plagued by a problem in which the speed of the CPU's operation exceeded the bandwidth of the memory subsystem to which it was connected. To avoid wasting CPU cycles while waiting for the memory to fetch the requested data, the universally adopted solution was to use an area of faster (and thus more expensive) memory to cache main memory data. This solution allowed the CPU to operate at its natural speed as long as the data it required was available in the cache.

The purpose of this article is to explain caching from the point of view of a kernel programmer. I also explain some of the common terms used to describe caches. This article is divided into sections whose kernel programming relevance is indicated; that is, some sections explain that cache properties are irrelevant to understanding the essentials of how the kernel handles caching. If you're coming from an Intel IA32 background, caching is practically transparent to you. In order to write kernel code that operates correctly on all the architectures Linux supports, however, you need to know the essentials of how caching works in general.

A Cache and Its Properties

Simply put, a cache is a place that buffers memory accesses and may have a copy of the data you are requesting. Usually one thinks of caches (there may be more than one) as being stacked; the CPU is at the top, followed by layers of one or more caches and then the main memory. In this hierarchy, caches are quantified by their level. The cache closest to the CPU is called level one, L1 for short, and caches increase in level until the main memory is reached.

A cache line is the smallest unit of memory that can be transferred to or from a cache. The essential elements that quantify a cache are called the read and write line widths. These signify the minimum amount of data the cache must read or write from the memory or cache below it. Frequently, these quantities are the same, so caches often are quantified simply by the line width. Even if they differ, the longest width usually is called the line width.

The next property that quantifies a cache is its size. This number is an indication of how much data could be stored in the cache. Often, the performance rule of thumb is the bigger the cache, the better the benchmarks.

A multilevel cache can be either inclusive or exclusive. Exclusive means a particular cache line may be present in exactly one of the cache levels and no more than one. Inclusive means the line may be present simultaneously in more than one level of cache. Nothing prevents the line widths from being different in differing cache levels.

Finally, a particular cache can be either write through or write back. Write through means the cache may store a copy of the data, but the write must be completed at the next level down before it can be signaled as complete to the layer above. Write back means a write may be considered complete as soon as the data is stored in the cache. For a write back cache, as long as the written data is not transmitted, the cache line is considered dirty, because it ultimately must be written out to the level below.

Cache Management and Coherency

One of the most basic problems with caches is coherency. A cache line is termed coherent when the data in the line is identical to the data stored in the main memory being cached. If this is not true, the cache line is termed incoherent. Lack of coherency can cause two particular problems. The first problem, which may occur for all caches, is stale data. In this situation, data has changed in main memory but the cache hasn't been updated to reflect the change. This usually manifests itself as an incorrect read, as illustrated in Figure 1. This is a transient error, because the correct data is sitting in main memory; the cache simply needs to be told to bring it in.

The second problem, which occurs only with write back caches, can cause actual destruction of data and is much more insidious. As illustrated in Figure 2, the data has been changed in memory, and it also has been changed separately by a CPU write to the cache. Because the cache must write out one line at a time, there now is no way to reconcile the changes—either the cache line must be purged without being written, losing the CPU's change, or the line must be written out, thus losing the changes made to main memory. All programmers must avoid reaching the point where data destruction becomes inevitable; they can do this through the judicious use of the various cache management APIs.

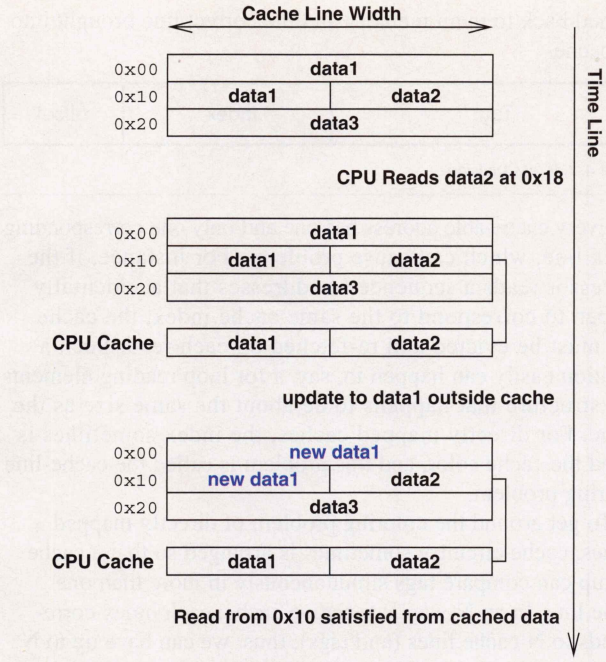


Figure 1. Stale Data Problem

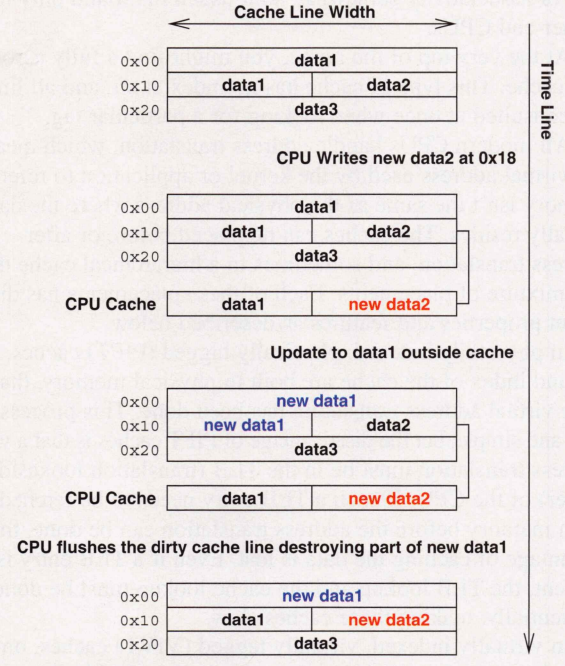


Figure 2. Data Destruction by Dirty Cache Lines

Cache-Line Interference

The situation where two sets of independent data lie in the same cache line, potentially leading to the data destruction detailed above, is termed *cache-line interference*. If you are laying out data structures in memory, the general rule to avoid this situation is never, ever have data that can be modified outside of the caches mixed with data the CPU may ordinarily use. If you absolutely have to violate this rule, make sure all externally modifiable elements of the structure are aligned *L1_CACHE_BYTES*, a value set at compile time to the largest possible cache width value for all the processors on which your code may run. The best thing to do is use kmalloc to allocate two separate regions. kmalloc never allocates two regions that overlap in a cache line.

Cache Management Instructions

The most basic instruction, called an invalidate, simply ejects the nominated line from all caches. Any reference to data in that line causes it to be re-fetched from main memory. Thus, the stale data problem may be resolved by invalidating the cache line before reading the data. In Linux, such an invalidation is done with:

```
void
dma_cache_inv(unsigned long address
              unsigned long size);
```

where *address* is the virtual address on which to begin, and *size* is the length of data to invalidate. Note that *size* is rounded up automatically to a multiple of the cache line width.

For write back caches, any dirty cache line may be written out, or flushed, to main memory using:

```
void
dma_cache_wback(unsigned long address,
                unsigned long size);
```

This flushing must be done before anything changes the

BIGGER RACKS.
LOWER T1
PRICING.

1U, 4U and 1/4 rack co-location services that can't be beat.

- OUR 1/4 RACKS ARE 15U (~26 INCHES)
- WE HAVE SOME OF THE LOWEST PRICES ON THE WEST COAST

1u to 2u $50/month
4u $75/month
1/4 rack $150/month

ALL PRICES INCLUDE BANDWIDTH

LAYER 42

WWW.LAYER42.NET
INFO@LAYER42.NET
408.450.5740

main memory under the dirty cache line. You therefore must issue the flush before an external entity (such as a PCI card) modifies main memory and issue an invalidate after this flush but before the CPU accesses any data that has changed.

In theory, for write back caches an invalidate kills the cache line without actually writing the data out, thus destroying the data in the cache. A safer thing to do in this case is issue a flush and invalidate instruction:

```
void
dma_cache_wback_inv(unsigned long address,
                    unsigned long size);
```

This flushes the cache lines to main memory and then invalidates them from the cache.

Types of Caches

This section explains how a cache actually works. The only vital piece of information you need from this section is a property called aliasing, which means that the same physical address in memory may be cached in multiple distinct cache lines. How the kernel actually manages the aliases is discussed in the following section.

In a directly mapped cache, as shown in Figure 3, the cache is divided up into cache lines of known width (four in the example). Each line in the cache is characterized by a unique index, so every byte in the cache is addressed by the index of the line and offset into the line. Each index of the cache also possesses a hidden number called the tag.

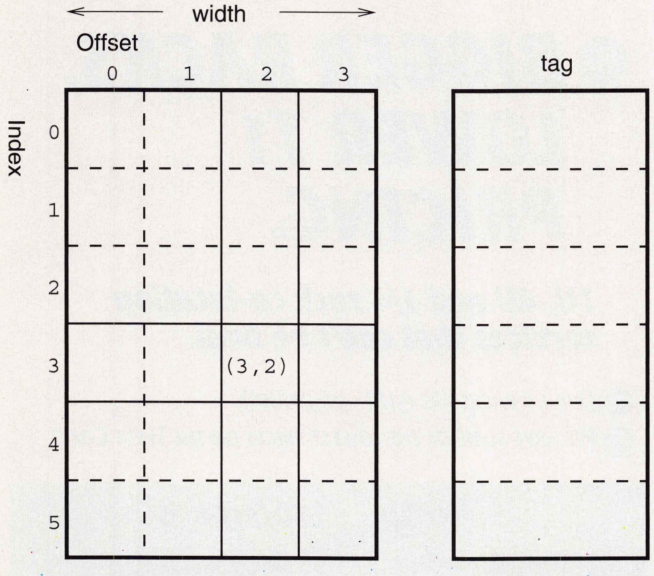


Figure 3. A Directly Mapped Cache

Every address in the system is divided into three pieces—tag, index and offset—along a power of two boundary (Figure 4). When a line is brought into the cache, the tag and index are extracted from the address. The line is stored in the cache at the required index, and the hidden tag is stored along with the line data. When the CPU makes reference to a particular address, the cache is consulted at the given index. If the tags match, the offset into the line is fetched to satisfy the address reference. If the tags do not match, the current line may be flushed back to main memory and the correct line brought into the cache.

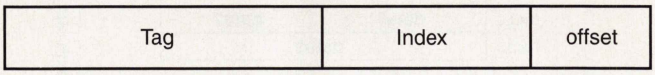


Figure 4. Address Division

Every cache-able address has one and only one corresponding index line, which can cause problems. For instance, if the processor reads a sequence of addresses that accidentally happen to correspond to the same cache index, the cache line must be evicted and re-fetched on each read. Such a situation easily can happen in, say, a for loop reading elements of a structure that happens to be about the same size as the cache. For directly mapped caches, the index sometimes is called the cache color, and this problem is called the cache-line coloring problem.

To get around the coloring problem of directly mapped caches, cache circuitry sometimes is arranged so that a cache lookup can compare tags simultaneously in more than one cache line. In an N-way associative cache, each index corresponds to N cache lines (and tags); thus, we can have up to N addresses with the same index simultaneously in the cache. The added parallel cache lookup circuitry tends to increase the cost of associativity somewhat, so it usually is found only in higher-end CPUs.

At the very top of the range, you might find a fully associative cache. This type of cache has no index at all, and all lines are consulted at once when looking for a particular tag.

All modern CPUs handle address translation, which means the virtual address used by the kernel or application to refer to memory isn't the same as the physical address where the data actually resides. The caches can be placed before or after address translation, and sometimes in a hierarchical cache there is a mixture of placements. Each of these placements has different properties and features as described below.

In physically indexed, physically tagged (PIPT) caches, the tag and index of the cache are both in physical memory, that is, after virtual address translation has been done. This process is nice and simple, but the disadvantage of PIPT caches is that a valid address translation must be in the TLB (translation lookaside buffer) of the CPU. If such a TLB entry needs to be fetched from memory before the address translation can be done, the advantage of caching the data is lost. Even if a TLB entry is present, the TLB lookup and the cache lookup must be done sequentially, making these caches slow.

In virtually indexed, virtually tagged (VIVT) caches, on the other hand, both the index and tag are in the virtual address space in which the CPU currently is operating. Although this makes cache lookups much faster (no address translation needs to be done before the lookup or after, if the cache lookup is successful), they suffer from several other problems:

1. Virtual address translations usually are changed as part of normal kernel operation, so the cache must pay careful attention to changes in TLB entries (and changes in address space) and flush cache lines whose translations have changed.

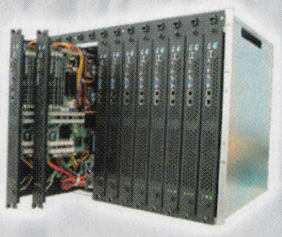
2. Even in a single address space, multiple virtual addresses may exist for the same physical address. Each of these virtual

VXRACK

THE FUTURE OF CLUSTER TECHNOLOGY

NEED TO **INCREASE PERFORMANCE WITHOUT** INCREASING **COST ?**

Get VXRACK™ Cluster Technology based on the **Intel® Xeon™ Processor** and **get the price**/performance leadership **you need**.



ONE RACK
1.5TFLPS
128 VXBLADE
1.5TB MEMORY
256 INTEL® XEON™

HOW BIG IS YOURS ???

*VX*RACK™
8 VXCubes
(up to 4 high and back-to-back)
128 VXBlade (VXB-7501)
256 Intel® Xeon™ Processors
More than 32 CPU per Sft
51 TB of Storage
1.5 TB of Memory
Patended Rack cooling system
Environmental Thermal Sensor
Dual Front End PDU (Optional)

*VX*CUBE™
Rack mounted Metal Chassis
16 VXBlades (VXB-7501)
32 Intel® Xeon™ Processors
16 or 32 Ports Patch Panel
(RJ45 or Infiniband Connectors
RS-232 In-band Connector)
Environmental Thermal Sensor
Integrated PDU (20Amp-208V)
Internal Gigabit Switch (Opt.)
Internal Topspin Switch (Opt.)

*VX*BLADE™ (VXB-7501)
VXBlade (VXB-7501)
Dual Intel® Xeon™ Processors
(Up to 3.06GHz)
Intel® E7501 Chipset
8 or 12GB of ECC DDR memory
Dual Extreme Active Heatsink
Dual Intel® Gigabit Ports
0, 1 or 2 IDE ATA 100 HDD
One PCI-X Slot Available
IPMI 1.5 Management (Opt.)

*VX*BLADE™ (VXB-8870)
VXBlade (VXB-8870)
Dual Intel® Itanium2™ Processors
(1GHz, 1.5MB L3 Cache)
Intel® E8870 Chipset
16GB of ECC DDR memory
Dual Extreme Active Heatsink
Dual Intel® Gigabit Ports
0 or 1 IDE ATA/100 HDD
One PCI-X Slot Available
IPMI 1.5 Management (Opt.)

AVAILABLE NOW...CALL US FOR MORE INFOS. **COMING 12/03**

AFFORDABILITY
VXRack Extreme Density coupled with Intel Server Building Block Components offer an incredible cost-benefit compared with regular 1U Cluster solution or proprietary RISC based Supercomputers

SCALABILITY
VXRack is inherently scalable. Increased processing loads can be quickly and easily accommodated either by adding identical VXBlade & VXCube, or by scaling up the nodes with faster processors or memory

EASY UPGRADE
VXRack is easy to upgrade as workload or budget allows. Each nodes can be upgraded without affecting the other. In addition, large clusters can be conveniently built in stages on pay-as-you-go basis

RAW PERFORMANCE
VXRack can be easily built to provide virtually any level of capacity and performance. A single VXRack with 256 Xeon Processors can deliver more than 1.5Tflp (Trillion floating point operations per sec.)

FLEXIBILITY
Portions of VXRack can be assigned to different applications, allowing several separate jobs to run efficiently in parallel, or consolidated to complete a single large execution as fast as possible

UNSURPASSED QUALITY
VXRack takes advantage of Intel's 20 years of experience designing and engineering industry leading server components. Intel spends 10,000+ hours testing and validating every piece of the server stack

intel inside XEON

intel inside ITANIUM 2

CALL TODAY 866-7VX-RACK OR WWW.VXRACK.COM

CIARA TECHNOLOGIES

Copyright © 2003, Hypertechnologies Ciara Inc. All Rights reserved. Hypertechnologies Ciara Inc. is not responsible for any typographical errors. Pricing, availability or pictures are subject to change without notice. Ciara Technologies, Ciara Technologies Logo, VXRACK, VXCUBE and VXBLADE are trademarks or registered trademarks of Hypertechnologies Ciara Inc.. Intel, the Intel Inside logo, Intel Xeon and Intel Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

addresses would be cached separately, even though they represent the same data. This is called the cache-line aliasing problem.

Generally, though, the added lookup speed of a VIVT cache outweighs its disadvantages, making it the predominant cache type for non-x86 CPUs.

A type of hybrid cache designed to overcome some of the shortcomings of the VIVT cache without sacrificing too much of its speed advantage is virtually indexed, physically tagged (VIPT) caching. The index is on the virtual address, but the tag is on the physical address, so the combination (tag, offset) must specify the full physical address. This requirement causes the tags to be larger than the tags for other cache types.

The VIPT cache gains its speed advantage over PIPT because the address translation and the cache lookup now can be done in parallel. The CPU doesn't know if the cache line is valid (the tags match), however, until the address translation has completed.

The disadvantages of VIVT are overcome because the tag is physical, thus the VIPT cache automatically detects aliasing when it sees that two tags are identical in the cache. Thus, a VIPT cache may be constructed in such a fashion that cache-line aliasing never occurs.

This fourth theoretical type of cache, physically indexed, virtually tagged (PIVT), is basically useless and is not discussed further.

The Aliasing Problem

Any time the kernel sets up more than one virtual mapping for the same physical page, cache-line aliasing may occur. The kernel is careful to avoid aliasing, so it usually occurs only in one particular instance: when the user mmaps a file. Here, the kernel has one virtual address for pages of the file in the page cache, and the user may have one or more different virtual addresses. This is possible because nothing prevents the user from mmaping the file at multiple locations.

When a file is mmaped, the kernel adds the mapping to one of the inode's lists: i_mmap, for maps that cannot change the underlying data, or i_mmap_shared, for maps that can change the file's data. The API for bringing the cache aliases of a page into sync is:

```
void flush_dcache_page(struct page *page);
```

This API must be called every time data on the page is altered by the kernel, and it should be called before reading data from the page if page->mapping->i_mmap_shared is not empty. In architecture-specific code, flush_dcache_page loops over the i_mmap_shared list and flushes the cache data. It then loops over the i_mmap list and invalidates it, thus bringing all the aliases into sync.

Separate Instruction and Data Caches

In their quest for efficiency, processors often have separate caches for the instructions they execute and the data on which they operate. Often, these caches are separate mechanisms, and a data write may not be seen by the instruction cache. This causes problems if you are trying to execute instructions you just wrote into memory, for example, during module loading or when using a trampoline. You must use the following API:

```
void
flush_icache_range(unsigned long start,
                   unsigned long end);
```

to ensure that the instructions are seen by the instruction cache prior to execution. *start* and *end* are the starting and ending addresses, respectively, of the block of memory you modified to contain your instructions.

General Cache Flushing

Two APIs globally flush the CPU caches:

```
void flush_cache_all(void);
```

and

```
void flush_cache_mm(struct mm_struct *mm);
```

These flush all the caches in the system and only the lines in the cache belonging to the particular process address space *mm*. Both of these are extremely expensive operations and should be used only when absolutely necessary.

Caching in SMP Environments

When more than one CPU is in the system, a level of caching usually exists that is unique to each CPU. Depending on the architecture, it may be the responsibility of the kernel to ensure that changes in the cache of one CPU become visible to the other CPUs. Fortunately, most CPUs handle this type of coherency problem in hardware. Even if they don't, as long as you follow the APIs listed in this article, you can maintain coherency across all the CPUs.

Conclusion

I hope I've given you a brief overview of how caches work and how the kernel manages them. The contents of this article should be sufficient for you to understand caching in most kernel programming situations you're likely to encounter. Be aware, however, that if you get deeply into the guts of kernel cache management (particularly in the architecture-specific code), you likely will come across concepts and APIs not discussed here.

James Bottomley is the software architect for SteelEye. He maintains the SCSI subsystem, the Linux Voyager port and the 53c700 driver. He also has made contributions to PA-RISC Linux development in the area of DMA/device model abstraction.

TECH TIP

If you've just converted all the GIF images on your Web site to PNG, but the HTML still refers to the original .gif filenames, just use Apache's mod_rewrite to redirect visitors to the new versions of the images:

```
RewriteEngine on
RewriteRule (.*)\.gif $1.png [R]
```

Multimedia

If last month's issue left you hungry for more audio and video projects, feast your eyes and ears on our Web site. **BY HEATHER MEAD**

Our new products mailbox has been receiving a lot of press releases introducing new workstations designed specifically for high-end audio and graphic needs. If you're an animator, filmmaker or a recording engineer who uses Linux, hardware and software companies are very interested in getting your business. But, even if you're a regular desktop user, Linux developments in audio and video matter, because at least some of them eventually will make their way into a desktop distribution. So this month, we're going to point you to some *LJ* Web articles that discuss what is going on in the Linux multimedia world.

If you're interested in a sound card that lets you do more complicated audio tasks, such as recording and mixing music, Peter Todd discusses "Using the Hammerfall HDSP on Linux" (www.linuxjournal.com/article/7024). The Hammerfall is a professional-grade sound card used in studios all over the world; it also happens to run under Linux, thanks to the ALSA Project drivers. It consists of an external module called the Multiface coupled to an internal PCI card. Peter explains that the big difference between the Hammerfall and regular sound cards is that the HDSP "is designed as a sound I/O device. It has inputs and outputs, and you can route sound between them arbitrarily." The article also explains how to use advanced features such as an external time code source, a must for the studio.

In the early spring of 2003, Dave Phillips, author of *The Linux Book of Music & Sound*, attended the first conference held specifically for Linux audio developers. The report he wrote for us, "Linux Audio Development: A Report from Karlsruhe" (www.linuxjournal.com/article/6762), provides an overview of the main issues discussed at the conference, as well as the directions Linux audio development is taking. Among the topics discussed was JACK, the software toolkit that "provides a professional-grade audio server in a low-latency environment, making arbitrary audio signal routing possible, without dropouts or distortion."

Moving on to the video portion of our presentation, Roberto de Leo's article "Self-Hosting Movies with MoviX" (www.linuxjournal.com/article/6474) explains how de Leo came to start the MoviX Project for a self-hosting movie—"a Linux CD mini-distribution that is able to boot and play automatically all audio/video files on the CD." The main point of the article, however, is to walk users through the process of building their own mini-distributions, whether for playing a movie or some other application.

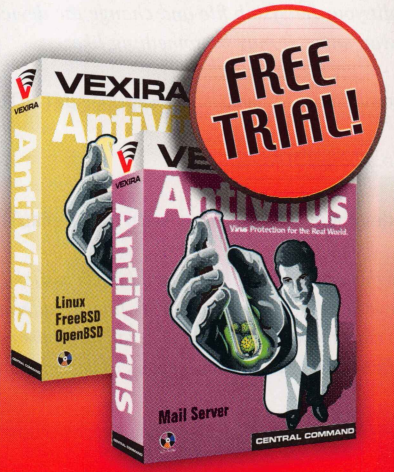
Finally, Geoff Draper wrote an article for us, "The Art of Rewriting Old Games" (www.linuxjournal.com/article/7083), in which he recounts how nostalgia for an old favorite, *Nellan Is Thirsty*, led him to rewrite the old 8-bit text game. The new version, *Thirsty Nellan*, "replaces the command-line parser interface of the original game with a point-and-click GUI environment." He also used Alias|Wavefront for the scenery, which we must admit, is some of the cutest penguin artwork we've ever seen. Because many of these old 8-bit games are in the public domain, they too can be re-created for another generation of gamers. Read the article to learn exactly what Draper did, and then go rescue your own old favorite.

If you do rewrite an old game, set up a music studio or edit films on your Linux machines, drop us a line and let us know how you're doing it. And, be sure to check the *Linux Journal* Web site often. New articles are posted daily.

Heather Mead is senior editor of *Linux Journal*.

Best of Breed

Virus Protection for Mail Servers



Vexira™ Antivirus for Mail servers is an industry leading, highly scalable virus scanning solution for email servers. Contact Central Command® today to find out why Vexira Antivirus is known as the "best of breed" in virus protection.

Supported OS:
- Linux
- FreeBSD
- OpenBSD

Email Servers:
- Sendmail
- Postfix
- Qmail
- Exim

Central Command, Inc.
phone: 330.723.2062 • fax: 330.722.6517
www.centralcommand.com

© 2003 Central Command, Inc. All rights reserved. Vexira, Vexira logo, and Central Command are trademarks or registered trademarks of Central Command, Inc. All other trademarks are property of there respective owner(s).

Fixing Swap When Moving a Drive

I am using Red Hat 9 on an HP Brio (PIII, 20GB and 192MB of RAM). I first installed Linux when my hard drive was a primary slave. Later I changed it to primary master. I am using a boot disk to boot in to Linux, but when I try to boot in to the system, I get an error that the swap partition was not initialized. The kernel could detect changes to my root and boot, but it was unable to in the case of swap partition. How can I change the swap partition dynamically? In other words, what parameter should I pass to the kernel before booting so as to specify my swap at the command line? Also, can I use the same boot disk to boot on other systems with different partition allocations, as with Windows startup disks?

--

Aman Hardikar, cybergeek2k@rediffmail.com

This is easy to explain; Linux is looking for the swap partition on one device (the primary slave), probably named /deb/hdbX, X being the partition number, and it happens that it now resides on the primary master disk, which probably is /dev/hdaX. Therefore, Linux can't find the designated swap partition and is unable to initialize it. To fix this, edit your /etc/fstab file and change the device on the line that has the swap entry. It may be something like:

```
/dev/hdb2    swap    swap    defaults        0 0
```

So, change it to:

```
/dev/hda2    swap    swap    defaults        0 0
```

In this example, the swap partition is number 2 on the disks. Then, reboot your system, and it should work fine. An alternate way to reboot would be to start the swap partition manually with the command `swapon -a -e`, *after editing the /etc/fstab file as indicated above.*

--

Felipe Barousse Boué, fbarousse@piensa.com

Serial ATA Support?

I intend to install SuSE 8.2 and want to know if I can install it on a Serial ATA hard drive.

--

Daniel Gustafsson, gustafsson_danie@hotmail.com

Serial ATA hard drives are supported by Linux (including the SuSE 8.2 distribution's stock kernel version 2.4.19). This will be obvious, but as always, first you must check that the BIOS and/or disk controller hardware of your system does support Serial ATA hard drives.

--

Felipe Barousse Boué, fbarousse@piensa.com

As usual, the first question is, does your BIOS support booting from your serial ATA interfaces? If your BIOS doesn't support booting from this controller, you might be able to install Linux to boot from some other device.

--

Jim Dennis, jimd@starshine.org

You can look up your hardware on SuSE's component database at hardwaredb.suse.de. *Other Linux distributions also maintain their own hardware compatibility lists. Although something technically might be supported in the kernel version that your distribution ships, it's best to check your distribution's hardware compatibility list before you go hardware shopping. Items on the list are more likely to be detected by the installer automatically, and your distribution is more likely to include up-to-date utilities with support for items on the list.*

--

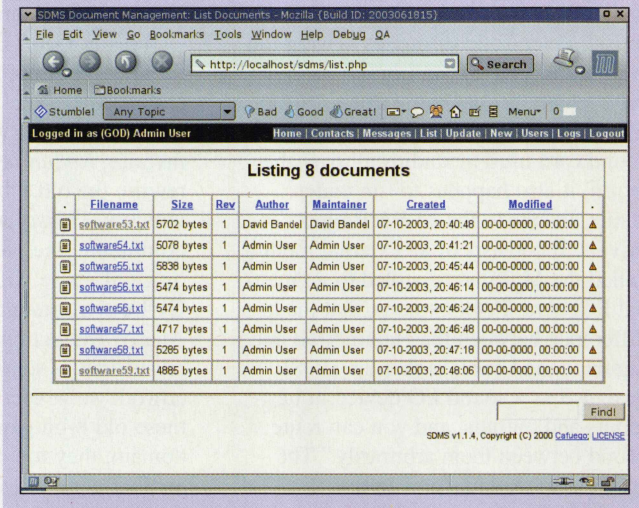
Don Marti, dmarti@ssc.com

Many on-line help resources are available on the *Linux Journal* Web pages. Sunsite mirror sites, FAQs and HOWTOs can all be found at **www.linuxjournal.com**.

Answers published in Best of Technical Support are provided by a team of Linux experts. If you would like to submit a question for consideration for use in this column, please fill out the Web form at **www.linuxjournal.com/lj-issues/techsup.html** or send e-mail with the subject line "BTS" to bts@ssc.com.

Please be sure to include your distribution, kernel version, any details that seem relevant and a full description of the problem.

Simple Document Management System:
sdms.cafuego.net



A lot of companies and even some individuals need a system to archive documents. This document management system uses MySQL for storage and provides varying levels of access to those who need it. Access can be denied, read-only, modify or god-mode (which allows almost anything) on a document-by-document basis. Requires: MySQL, PHP, Web server and Web browser.

—DAVID A. BANDEL

OpenOffice.org 1.1

OpenOffice.org has released version 1.1 of its office suite for Linux, Windows and Solaris in a variety of languages, including English, German, Swedish, Spanish and Japanese. New features for version 1.1 include native, one-click PDF export, Flash export, a faster load time, improved MS Office file compatibility and accessibility support. In addition, support for AportisDoc, Pocket Word and Pocket Excel has been added for easier transfer of documents to PDAs and handheld devices. OpenOffice.org 1.1 also offers an open, future-proof XML file format, which allows users to avoid file format lock-in situations. The suite and its source code can be downloaded from www.openoffice.org/dev_docs/source/download.html.

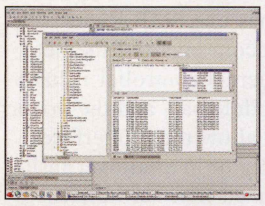
CONTACT OpenOffice.org, www.openoffice.org.

Panasas ActiveScale Storage Cluster

Employing an object-based architecture that manages data as large virtual objects, the ActiveScale Storage Cluster is comprised of the ActiveScale File System, StorageBlades and DirectorBlades, a 4U system shelf and an integrated gigabit Ethernet switch. The ActiveScale File System turns files into smart data objects and then dynamically distributes data activity across the StorageBlades. This design enables parallel data paths between servers and StorageBlades, eliminating performance and capacity bottlenecks. Application and user data is stored in the StorageBlades, and filesystem activity is coordinated by the DirectorBlades. DirectorBlades also virtualize data objects across the range of StorageBlades so the system can be viewed as a single namespace.

CONTACT Panasas, 6520 Kaiser Drive, Fremont, California 94555, 510-608-7790, www.panasas.com.

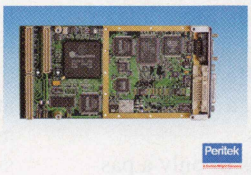
Aqua Data Studio 3.5



Aqua Data Studio 3.5 is a database query and administration tool that allows developers to create, edit and execute SQL scripts, as well as browse and modify database structures. Aqua Data Studio provides an integrated database environment with a single consistent interface to all major relational databases. This allows the database administrator or developer to tackle multiple tasks simultaneously from one application. Aqua Data Studio includes support for such database platforms as Oracle 8i/9i, IBM DB2, Informix Dynamic Server, Sybase Adaptive Server, Microsoft SQL Server, MySQL and PostgreSQL. New features for version 3.5 include complete schema extractions, SQL scripting of database objects and expanded Query Analyzer options.

CONTACT AquaFold, Inc., www.aquafold.com.

Stratus/PMC



The Stratus/PMC is a dual-head graphics controller PMC card designed for use in VME and CompactPCI embedded systems. It features a 128-bit graphics module for video capture with analog/digital I/O. Stratus/PMC's two display channels simultaneously deliver dual VGA or LVDS outputs or a single DVI output. Using the Silicon Motion SM731 graphics accelerator, Stratus/PMC supports single displays at screen resolutions up to 1600×1200×24bpp (with 16.7 million colors) and dual displays at 1024×768×24bpp. It also has an onboard 33/66MHz, 32-bit PCI interface; 16MB of SDRAM on-chip; composite and S-Video PAL/NTSC/SECAM support; and field-programmable VGA BIOS EEPROM. Stratus/PMC offers support for VxWorks, XFree86 and Microsoft Windows.

CONTACT Peritek, 5550 Redwood Road, Oakland, California 94619, 800-281-4567, www.peritek.com.

Xybernaut ServicePoint

Xybernaut has announced the release of ServicePoint, a mobile mini-server that is configured and shipped as a mobile application services platform. ServicePoint can function as a mobile or even wearable mini-server to direct information and application services to various devices and users over a wireless or wired network. Shipping with Red Hat Linux Professional Server, ServicePoint devices include a 500MHz Mobile Celeron Ultra Low Voltage Memory and Storage processor, 256MB of SDRAM, 5 or 10GB internal expandable hard drives, optional external drive ports, CompactFlash and USB ports, 8MB of SDRAM and a built-in sound card for full-duplex, stereo I/O.

CONTACT Xybernaut Corporation, 12701 Fair Lakes Circle, Suite 550, Fairfax, Virginia 22033, 703-631-6925, www.xybernaut.com.

GEME Series Embedded Computers

Adlink's new embedded computer series, the General Embedded Machine Engine (GEME), is designed for measurement, automation and communication needs. GEME combines an embedded SBC and power supply unit with optional storage peripherals, such as CompactFlash or a 2.5" hard drive. Featuring multiple mounting schemes, the modular chassis design allows one PMC and multiple PC/104 modules to be added. GEMEs come in three base configurations, all featuring low voltage, fanless processors and various extension modules, including video capture, gigabit Ethernet interface, remote I/O, serial communications and 64-channel DI/DO.

CONTACT Adlink Technology America, Inc., 15279 Alton Pkwy., Suite 400, Irvine, California 92618, 949-727-2077, www.adlink.com.

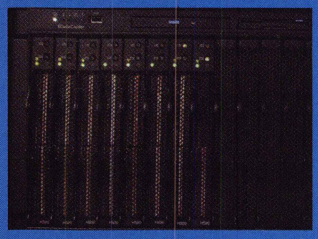
NemeSys DAW64

RackSaver and AMD have collaborated on the new NemeSys 64-Bit Digital Audio Workstation (DAW64), which features AMD's 64-bit Opteron processor. Designed for digital audio processing, editing and mastering needs in a studio environment, DAW64 is a 4U rackmountable system with dual Opteron processors, 4GB of RAM, four 36GB Raptor SATA hard drives, a 3ware RAID controller and an NVIDIA Quadro 4 380XGL graphics card. The 64-bit address spaces provide users with direct access to virtual instruments and music libraries in physical memory, eliminating the need for distributed content on multiple workstations.

CONTACT RackSaver, Inc., 9449 Carroll Park Drive, San Diego, California 92121, 858-874-3800, www.racksaver.com.

Please send information about releases of Linux-related products to Heather Mead at newproducts@ssc.com or New Products c/o *Linux Journal*, PO Box 55549, Seattle, WA 98155-0549, or fax to 206-782-7191. Submissions are edited for length and content.

IBM eServer BladeCenter



REVIEWED BY DANA CANFIELD

PRODUCT INFORMATION

Manufacturer:
IBM

URL:
www.ibm.com/servers/eserver/bladecenter

Price:
BladeCenter Chassis from $2,789 US; HS20 Blade from $1,879 US; Ethernet Switch Module $2,199 US

THE GOOD

- 14 servers/28 CPUs in 7U rack space.
- Fully redundant design throughout the system.
- Full Linux support on par with Windows support.
- Price point matches standalone servers after seven blades.

THE BAD

- Only one console for 14 systems.
- Support limited to certain versions of Red Hat and SuSE.
- Limited expandability.

IBM recently has begun rallying around two major server systems for Linux deployment on the x86 architecture. On the scale-up front, IBM offers the eServer x440 line, which boasts scalability up to 16 CPUs in a single system. More interesting to the average user with a cluster of smaller servers is IBM's scale-out offering, the IBM eServer BladeCenter system, a 7U rackmount chassis supporting 14 blades.

The chassis itself consists of only a passive midplane, a CD-ROM and floppy drive and two 10" blowers. Two redundant 230V power supply modules ship with the unit, and two more are required if the system is running more than five blades. A single management module with keyboard, video and mouse (KVM) and Ethernet connections is included, and support for a redundant management module recently was added.

Redundancy is a key design goal of the BladeCenter and little has been overlooked. Four switch-module bays are in the rear of the chassis; the second and fourth bays are redundant bays for the first and third bays, respectively. For example, if you install an Ethernet switch in the first bay, you can install a second Ethernet switch module in the third bay but not a SAN switch module. Although allowing full redundancy, this means that only two types of switch modules ever can be installed at a time.

Currently, IBM offers gigabit Ethernet and Fibre Channel SAN switch modules. Internally, these switch modules connect to each individual blade, and externally the gigabit switch offers four copper Ethernet ports while the SAN switch offers four Fibre Channel ports. With redundant modules installed, each blade has access to two Ethernet ports and two SAN ports. The Ethernet switch is a full-function layer-3 switch with a powerful Web-based interface. The SAN switch is a Q-logic SAN switch, which is configured through either a telnet session or the Java-based management software available for both Linux and Microsoft Windows.

The front of the BladeCenter chassis contains a single CD-ROM, floppy and USB port that is connected by an internal USB hub/switch to each blade. This switch is controlled by a button on each blade, and only one blade can access the CD/floppy/USB port at a time. The keyboard, video and mouse are connected through a separate internal switch that is controlled by a second button on each blade. Control of the KVM and USB peripherals also can be manipulated by using the BladeCenter's Web-based management interface, which allows the KVM output to be controlled remotely over the Web. Unfortunately, although redundant KVM/management modules may be installed, only one blade can be assigned to the console at a time, even through the Web interface. This probably is the most disappointing limitation of the BladeCenter, but it may be less of an issue for those using the blades in a true cluster environment.

At the time of this writing, only the HS20 server blade has been released for the BladeCenter. One might assume these blades simply would be a modified ThinkPad on a card, but in reality the blade is a miniaturized, enterprise-class server. Although it ships with a single CPU and 512MB of RAM, the HS20 has a 533MHz front side bus and can support two Pentium 4 Xeon CPUs (up to 2.8GHz) and four PC2100 DDR DIMMs (up to a total of 8GB). Thanks to the hypertheading feature of the Xeon CPUs, modern Linux kernels can see two logical CPUs for each actual CPU installed.

Each blade also has one 8MB Rage Video and two Broadcom gigabit Ethernet controllers. Each Ethernet controller is hardwired to a switch-module bay, and the driver can be configured to allow these interfaces to be used individually or in a failover fashion. Two notebook-sized IDE hard drives can be mounted in the blade, but the optional

One might assume these blades simply would be a modified ThinkPad on a card, but in reality the blade is a miniaturized, enterprise-class server.

SAN controller card overlaps with the second IDE drive, so you can use only one IDE drive if you also are connecting to a SAN. The SAN card is a custom Qlogic SAN card, which provides redundant pathways to the SAN switch modules. IBM also offers a companion blade that allows two SCSI drives to be attached to a blade, but it consumes another slot in the chassis.

The blades are hot-swappable and generally easy to upgrade. If a blade suffers catastrophic failure, the blade can be removed and an onboard diagnostic system flashes an LED next to the faulty component. I did find that installing the heatsink on the second CPU required a disturbing amount of force due to the spring-loaded screws used to secure it.

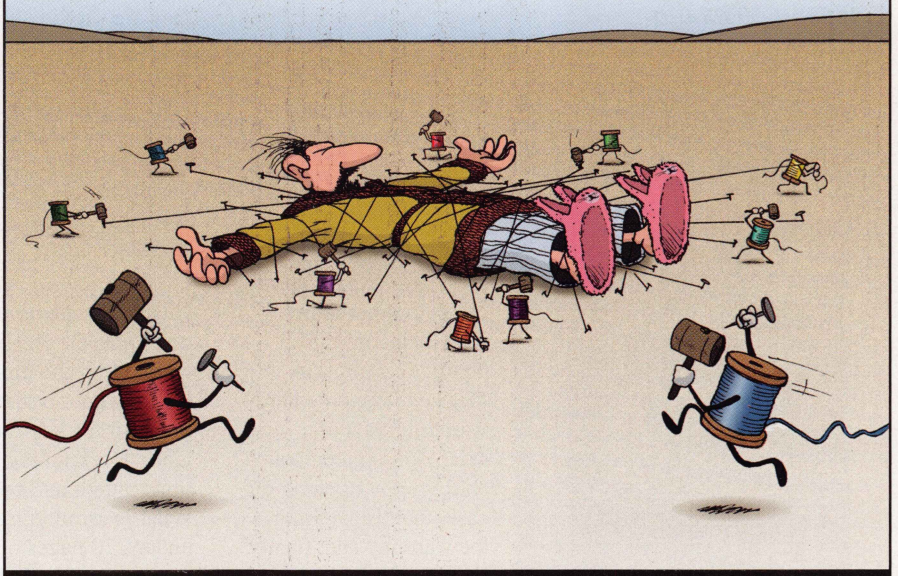
An interesting design characteristic of the BladeCenter is that IBM promises they eventually will release blades based on their pSeries and iSeries server lines. This means you can mix and match both Intel and mainframe-class blades in the same chassis. At the time of this writing, no such products have been announced formally, though.

As mentioned earlier, the BladeCenter has a powerful Web-based interface. Blades can be powered on and off, the CD/floppy or console port can be switched and firmware can be updated over the Web. At this time, user names and passwords must be defined within the management console, which is bothersome because those user names then must be defined again in each Ethernet and SAN switch module. In addition, security controls are coarse. Users cannot be granted access to only certain blades or limited to performing only certain tasks. Although users can be limited to having only view access, this limit also prevents those users from utilizing the remote console.

In addition to the Web interface, the BladeCenter includes licenses for the powerful IBM Director 4.1 management software, which is so featureful it deserves its own review. It is worth mentioning that the client, agent and server components of Director 4.1 all run under Linux and allow you to perform diagnostics, start and stop processes and receive automated alerts about all IBM servers from a single console.

Support for Linux on the BladeCenter is far better than I have found from any other server vendor. Linux driver CDs are supplied in the box, and all general management tools are written in Java with complete

CODY WAS QUICKLY OVERWHELMED



Characters and Images © 2003 Brad Fitzpatrick, ActiveEdge. All Rights Reserved.

Etnus TotalView... *The Best Thread Debugger on the Planet*

Switch threads with a single click, control a single thread while leaving all others in their current state, define and control thread groups, and much more. Designed from the ground up to handle multiple streams of execution, TotalView gives you what you need to debug even the most complex threaded programs. Don't get tied down by threads.

Try TotalView FREE at www.etnus.com.

ETNUS TOTALVIEW

Linux support. In fact, I have yet to find any management, monitoring or configuration tools related to the BladeCenter that are Windows-only. This is a refreshing change from past experiences with even the most dedicated Linux hardware vendors. Although official support currently is limited to Red Hat Linux 7.3, Red Hat Advanced Server 2.1 and SuSE Linux 8, I had no trouble running Red Hat 8.0 and 9.

If you have need of any custom peripherals, cards or interfaces on your server, the BladeCenter is not an option as there are no PCI or per-blade I/O ports. The addition of a single USB port dedicated to each blade would have made the BladeCenter a more versatile system.

Another important consideration is that if you do not have a dedicated server room, the BladeCenter can be uncomfortably loud. In fact, it is so loud that IBM sells an Acoustic Attenuation Unit (AAU), which essentially is a large foam muffler. Even with the AAU in place, it is possible to hear the BladeCenter running through an office wall.

IBM claims the BladeCenter is priced so that it is a lower cost alternative to standalone servers when buying seven or more blades. It is uncommon to find standalone servers with the level of redundancy that the BladeCenter includes, but I believe the cost claim to be true even compared to servers with fewer features. Overall, the BladeCenter is a remarkable bit of engineering. It is probably best suited for those building high-density clusters, but when coupled with a SAN, it makes a great option for general-purpose servers as well.

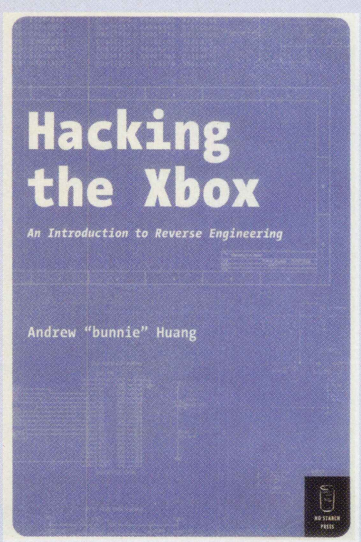
Dana Canfield is a systems administrator at the University of Indianapolis, where he constantly nags vendors to provide Linux support for anything and everything. He can be reached at canfield@peng1.net.

BOOK REVIEW

Hacking the Xbox

by Andrew "bunnie" Huang

No Starch Press, 2003
ISBN: 1-59327-029-1
$24.99



Structured around a series of hardware hacking projects that range in complexity from installing an alternate LED and making cables to eavesdropping on the motherboard with a custom tap board, *Hacking the Xbox* explores the hardware internals of Microsoft's game console first by opening the case, then by voiding the warranty and, finally, by taking a look around. Doing so is a legal minefield under the controversial Digital Millennium Copyright Act (DMCA), and discussion of the DMCA is featured frequently throughout the text. Although primarily of interest to the North American reader, the book provides insight into the problems that can be created by well-intentioned but often ill-informed legislators. Hackers living in other countries may not worry too much about the DMCA, but that's not to say there is not an equivalent local legislation waiting in the wings.

Readers looking to learn about hacking the Xbox from a software perspective will be intrigued but disappointed. Although it describes some important Xbox software projects, *Hacking the Xbox* focuses on reverse engineering from the point of view of the hardware engineer, not the software engineer. Detailed instructions on getting GNU/Linux up and running on your Xbox are not here, but they are provided elsewhere, and Andrew Huang includes the required pointers. If you are the kind of person who is at home with a screwdriver, a soldering iron, electronic components and PCBs, then this is the book for you. Of course, it helps if you can stomach potentially destroying your Xbox in the process. Not to worry, though, as Huang provides detailed instructions on replacing damaged mass-storage devices, memory and power supplies should anything go wrong.

Subtitled *An Introduction to Reverse Engineering*, Huang's *Hacking the Xbox* is much more. It is an introduction to cryptography and its use in hardware design. It is a warning on the threat to scientific pursuit that is the DMCA. It is a primer on intellectual property law and its implications for hackers. It is a profile of the key players in the Xbox Hacking community and it highlights their motivations, intent and contributions. It's also a darned good read. What is astounding is that Huang manages to do all this in under 270 pages.

Obviously, *Hacking the Xbox* describes the current Xbox and, now that the book has been published (another story all in itself), it is likely that the next-generation Xbox will change, invalidating the bulk of the book's content. That said, the general hardware hacking techniques described in *Hacking the Xbox* can be re-applied to the new Xbox, albeit without the step-by-step instructions. Of course, with millions of original Xbox consoles shipped worldwide, there will always be old, replaced or upgraded Xboxes to play with, so the book will have worth for some time to come.

—PAUL BARRY

Linux, Talon and Astronomy

A new open-source application lets professional and amateur astronomers explore space from their desktops. **BY TONY STEIDLER-DENNISON**

We've been fascinated by astronomy since the ancient Chinese first charted the skies. The computerization of astronomy, some would argue, is its greatest leap forward yet. Today, unattended robotic telescopes scan skies that have been charted over centuries, recording their findings in modern databases. CCD cameras capture images impossible to define on film. It's an exciting time to be an astronomer, whether amateur or professional.

The revolution in astronomy doesn't stop at the hardware. Research-grade telescopes in observatories from Spain to Korea are under the control of open-source software and Linux-based computers. Under the open-source model, scientists are free to modify the control software, creating a trickle-down effect that benefits amateurs. Open source and Linux even have changed the scientific method. With source code freely available, peer review now occurs not only on the data, but on the data gathering methods as well.

At the forefront of this open-source astronomy revolution is Talon. Talon was originally developed by Ellwood Downey as the Observatory Control and Astronomical Analysis Software (OCAAS). In 2001, the software was purchased by Torus Technologies of Iowa City, Iowa. In late 2002, Torus was purchased by Optical Mechanics, Inc., and the updated OCAAS package was released as Talon under the GPL.

During the past two years, I've had the daily pleasure of working with Talon. I've installed and configured the software on multiple telescope packages, and I've followed these telescopes to destinations around the world for installation and on-site configuration. It's my pleasure to share with you some of the broad points of Talon installation, configuration and use.

Talon can be downloaded at observatory.sourceforge.net. The software interacts with integrated motion control boards, available from Optical Mechanics, Inc. (Optical Mechanics Motion Controllers) or Oregon Microsystems (PC39 Motion Controllers). Object acquisition and tracking, scheduled operations, environmental monitoring, dome control, image analysis and processing all fall under the control of Talon. Networked operations also are possible using a remote X session.

Using Talon

The Talon package contains a full installation script; install.sh creates a talon user, compiles the binaries and creates a set of text configuration files for initial operation of the telescope.

Talon contains a full compliment of astronomy applications designed specifically for use as a suite of tools. The main Talon interface utilizes the Motif toolset, producing a familiar and unified look and feel throughout the application set. Although the toolset is rich, the following four tools should be of use to most observers.

xobs is the main Talon control window and is launched with the terminal command `startTel`. It contains all the monitoring and calibration tools necessary for operation. This window provides manual control of the telescope and any attached peripherals, such as a filter wheel or dome control. It also provides a constant display of the current position of the telescope, as calculated by feedback from the motor encoders. This feedback is provided in a set of text boxes within the xobs window.

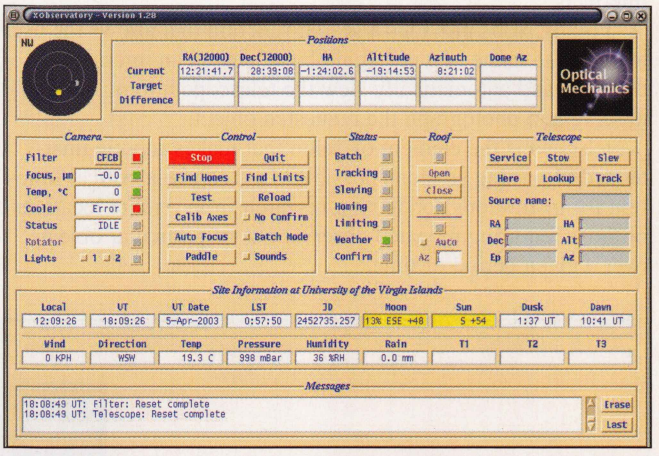


Figure 1. xobs, the Main Talon Control Screen

When using Talon, the first important task is to find the home position of the various encoders located throughout the system. These encoders close the loop on the operation of the axes, providing a static count for the full travel of each axis. Movement of the telescope is calculated in part by the motion of the chosen axis in relation to the zero position on the encoder. Decrementing the Declination encoder, for example, generally moves the telescope to the north. The operation to find homes in the xobs window hunts for and establishes the zero positions on each axis encoder.

Using the software paddle command in the xobs window (Figure 2), the user can position the telescope, filter wheel and focus position manually. The motion of the telescope to the east and west is referred to as the right ascension (RA) or hour angle (HA) of the telescope. To travel north and south is referred to as Declination (Dec). Using positive and negative encoder counts, moving the telescope axes is a simple matter of moving the axis positive or negative x (RA) or y (Dec). These coordinates are in relation to the North Pole.

Additionally, Talon provides data on the weather conditions at the observing site with an attached Davis weather station. This feature ensures that the telescope is not exposed to adverse weather conditions during unattended operations. When conditions fall within a predetermined range, the observatory dome or roll-off roof closes, the telescope moves to a stowed position and operations cease. As with the position

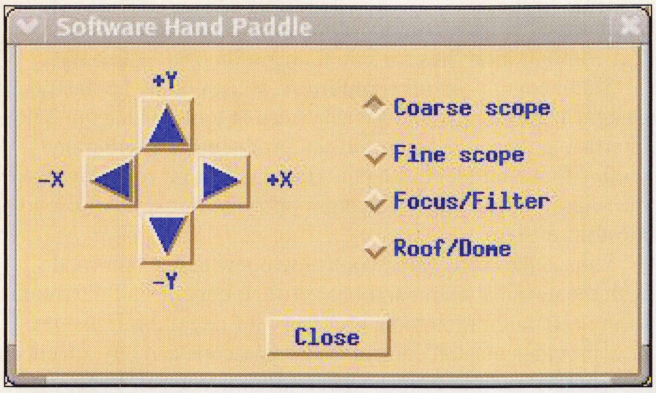


Figure 2. The Talon Software Paddle, Used to Move the Telescope Manually

data, this information is provided in text boxes within the xobs interface.

Finally, xobs provides a search function that allows the user to enter the name of a celestial object, search an internal database and automatically slew the telescope into position to observe and photograph the requested object.

telsched is the element of Talon that makes robotic unattended observing sessions possible. This can be a critical function for institutions conducting research from remote locations or those requiring repeated observations of particular objects over a given period of time.

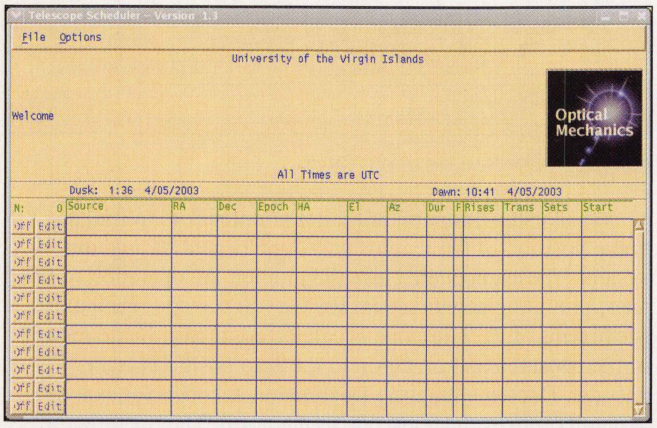


Figure 3. telsched, the Talon Operations Scheduling Program

The telsched command opens a scheduler for these unattended observing sessions. The scheduler automatically calculates images to be taken during the session based on the size (in degrees) of the chunk of sky the user selects. In general, the tighter the area of the sky (fewer degrees), the more images taken. Images taken by telsched during an unattended session are stored in a directory of the user's choice. All instructions created by the telsched program are stored in a flat file. These instructions are referenced by xobs when the telescope is slaved off to robotic control from the xobs interface.

Camera is another terminal-launched application in the Talon suite. It provides complete control over the functions of a CCD camera attached to the telescope.

The camera application includes tools for exposure time, image size, software image filtering and image analysis. Camera also contains tools for adjusting the brightness and

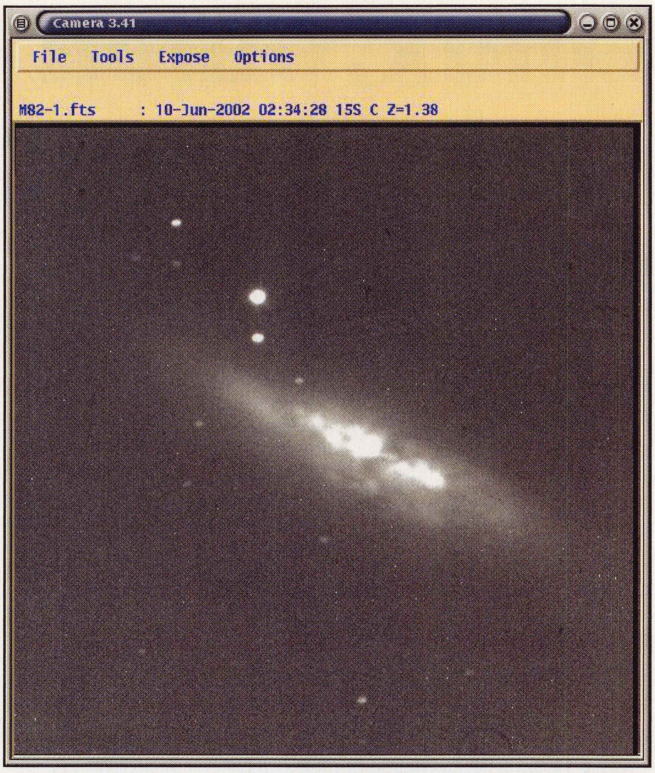


Figure 4. Camera, the Talon Image Processing Application

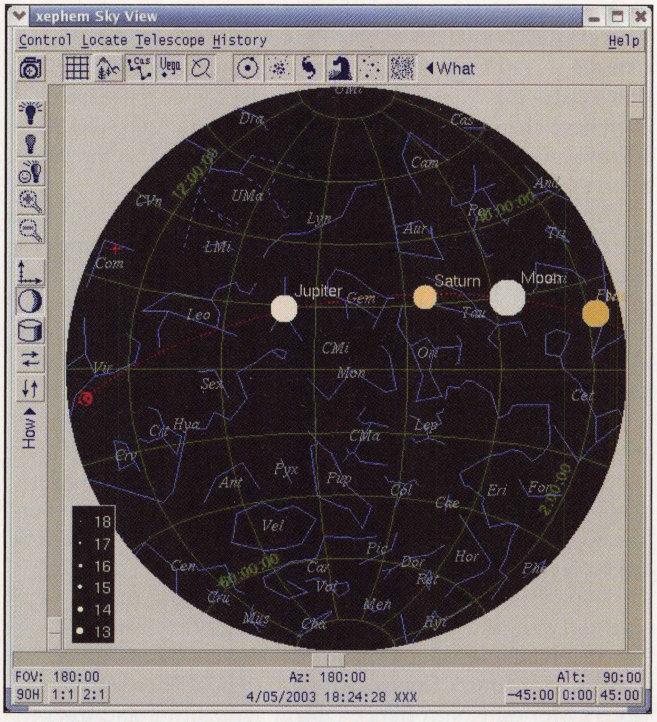


Figure 5. xephem, the Ephemeris Program for Talon

contrast of images, determining the area of interest (AOI) of the image and automatically labeling objects by comparison to the World Coordinates System (WCS). The latter tool is, in fact, a pattern-matching algorithm that allows the system to compare known patterns of objects to the WCS database.

xephem provides a software ephemeris, or sky charting interface, for the rest of the Talon suite. As with other ephemerides, it relies heavily on correct geographical and time coordinates; this information can be configured manually by the user. xephem also can be configured to poll an attached GPS at regular intervals, adjusting the system time to account for internal clock drift.

The xephem program, launched from the command line with `xephem`, provides a granular view of the current sky. Data on each object is provided in a right-click pop-up screen. The user also can point the telescope using this pop-up, a feature used extensively for calibration. Magnification can be increased, effectively looking deeper and deeper into the sky. As an alternative to zooming, the user may select a minimum magnitude (apparent brightness) threshold. This allows brighter stars to be filtered in the ephemeris view, leaving only the dimmer objects in the window. The sky view also may be rotated, and object type filtering is provided. For example, globular clusters can be selected, eliminating the view of all other object types.

Configuration

Configuration files are critical to the operation of Talon. They provide the means by which the software communicates with both the user and the hardware installed in the telescope. In the Linux tradition, these files are simple text files, commented heavily for the clarification of the user. The configuration files for all elements of Talon can be found in /usr/local/telescope/archive/config. Using the default tcsh shell, the simple command `cd config` moves the user into the configuration directory.

The operation of the telescope can be viewed as two discrete elements, each of which is addressed by a specific configuration file type. First, the internal motion control boards must communicate with the motors and encoders. Configuration files intended to serve this function utilize a .cmc extension. I've always viewed this extension as delineating files that configure motion controllers, cmc for short. The .cmc files establish the operating parameters for the controller boards, which, in turn, send signals to and receive feedback from the encoders and electromechanical components.

The other element of telescope operation is the interface between the user and the software. In simple terms, all user-controlled operations utilize configuration files with a more traditional .cfg extension. Whereas the .cmc files operate behind the scenes to communicate directly with the hardware, the user interface must communicate with the .cfg files.

Critical Configuration Files

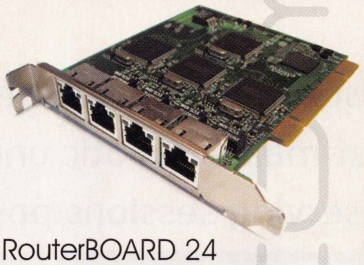
Although every configuration file plays a role in the operation of Talon, some in both the .cmc class and the .cfg class bear special attention. These .cmc files include:

- basic.cmc: establishes the basic communication between the motion control boards and the motors driving the telescope axes.
- find.cmc: establishes the routines for finding objects based on encoder counts.
- nodeDec.cmc: establishes the hardware parameters for the Dec axis of the telescope.
- nodeRA.cmc: establishes the hardware parameters for the RA axis of the telescope.
- nodeFocus.cmc: establishes the hardware parameters for the telescope focus control.

The .cfg files are:

- boot.cfg: allows the user to script Talon startup routines. These may include starting GPS monitoring, weather station monitoring and opening the Talon main interface when the computer boots.
- home.cfg: provides an initial set of constants to allow the telescope to find the home position of each encoder. These constants represent a spatial sense for the telescope prior to working through the initial calibration routines. Once these routines are completed, the actual encoder counts and axis travel are updated automatically.

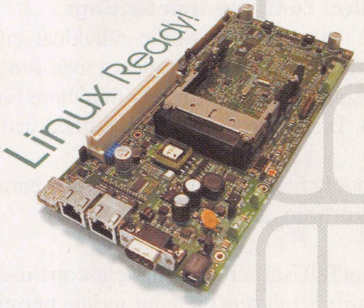
expand! $125



RouterBOARD 24

Four ethernet ports in one card that work as separate interfaces. Supported by all OSs!

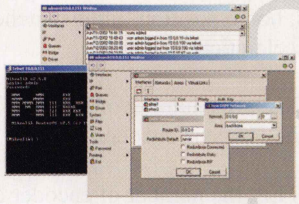
combine! $195



RouterBOARD 230

A super small integrated motherboard with various interfaces and 266 MHz CPU. Use as a CPE or an AP on a tower with 500ft of PoE (802.3af),or anything else you can imagine! Many cases and accessories available. IDE/CF, PCMCIA, miniPCI, PCI, USB.

route! $45



MikroTik.com RouterOS

A router operating system that has every possible feature a router could need - works with regular pc's and routerboards!

more!

Four-port miniPCI to PCI $95
One-port miniPCI to PCI $33
v.35 synchronous miniPCI $550
and much more on ...

RouterBOARD.com

sales@routerboard.com

RouterBOARD, RouterOS, and MikroTik are trademarks of Mikrotikls SIA

telsched is the element of Talon that makes robotic unattended observing sessions possible.

- telescoped.cfg: provides constants regarding the telescope axes, establishes the position of physical travel limit switches in relation to the encoders and establishes the maximum rotational velocity of each axis as well as the rotational acceleration rates.

Critical Configuration Settings

The settings in each of the individual .cmc and .cfg files utilize a naming convention that makes their function easily recognizable, but some critical settings within these files deserve special attention. These settings can be modified with any familiar text editor:

- boot.cfg: establishes the overall parameters of the Talon software at boot.
- setTelUser: creates the telescope user, the telescope user group and sets the appropriate permissions. By default, the initial telescope user and group are named talon. This can be changed for subsequent use by modifying the setTelUser constant in boot.cfg, provided the new user and group already exist on the system.
- setTelDaemons: initializes the telescope dæmon (telescoped), camera dæmon (camerad), weather station dæmon (wxd) and global positioning system dæmon (gpsd).
- home.cfg: provides the following four constants for encoder counts, home position, limit switches and rotational velocity and acceleration:
  - HSTEP: the number of encoder counts in the full rotation of the HA axis encoder.
  - DSTEP: the number of encoder counts in the full rotation of the Dec axis encoder.
  - HSIGN: the physical location of the HA encoder on the telescope. When viewed from the north, the HA encoder will increment clockwise if placed at the back of the polar shaft (the shaft upon which the telescope moves from east to west) or decrement when placed at the front. Another way to view this is, if the marked encoder surface points to the south in the final telescope configuration, it will increment when rotating clockwise. If it points to the north, the encoder will decrement with clockwise rotation. This configuration is a simple constant: 1 if the encoder increments, -1 if it decrements.
  - DSIGN: the physical location of the Dec encoder on the telescope. Much like the HA encoder, the increment/decrement of the encoder varies depending on the method used to mount the encoder. If the encoder is installed with the encoded surface toward the outside of the fork, it decrements when rotated clockwise, or toward the north. This requires a setting of 1. If the encoder is mounted with the encoded surface to the inside of the fork, it increments when rotated clockwise. This requires a setting of -1.
- telescoped.cfg: provides the following constants for initial operation:
  - HAXIS: the telescope network node from which the HA axis is controlled.
  - DAXIS: the telescope network node from which the Dec axis is controlled.
  - HESTEP: the raw encoder counts per revolution for the HA axis.
  - DESTEP: the raw encoder counts per revolution for the Dec axis.
  - HMAXVEL: the maximum slewing velocity of the HA axis.
  - DMAXVEL: the maximum slewing velocity of the Dec axis.
  - HMAXACC: the maximum slewing acceleration of the HA axis.
  - DMAXACC: the maximum slewing acceleration of the Dec axis.

Using Talon

Putting Talon to use requires a few initial calibration items. As with any telescope, you'll need to check and adjust the polar alignment—the physical location of the telescope in relation to celestial north.

With xobs in the boot.cfg script, the main Talon screen should open right after your desktop loads. From this main screen, select Find Homes (Figure 6). As noted, this routine finds the home mark on each encoder, RA, HA and Focus. From the pop-up window, select All. The telescope should move in all axes. Each axis skips past the home mark initially, backing up incrementally until it

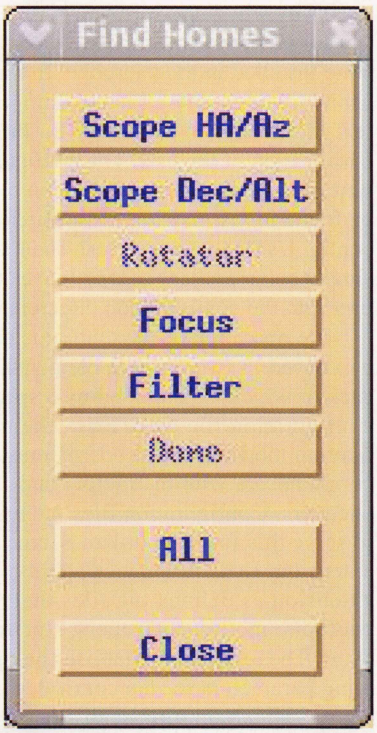


Figure 6. Finding Home

finds the mark again. This reduction of each move to the home mark ensures that the telescope ends up precisely on the mark.

The next step is to find limits. This routine locates the telescope's physical limit switches, which prevent the telescope from damaging itself by swinging too far through the travel of each axis. When the switches at both ends of travel are found in an axis, the software writes the location (in encoder counts) to the home.cfg file. You should need to complete the find limits routine only one time.

With the telescope calibrated and aligned, it's time to take some pictures. Open the camera and xephem applications from the command line. Enable telescope control in the xephem options. Select an object by right-clicking in the ephemeris, then select Point Telescope from the resulting pop-up. The telescope should slew to the new position. Click on the camera application and select Take One. With the proper connection to the camera, you'll hear the shutter trip. Within a few seconds, an image of the selected object renders on your screen.

To set up scheduled operations, use the telsched command. In the resulting window, select the size of the mesh, remembering that the tighter the mesh is, the more images taken. Set the time for the operations to start (in UT) and save the schedule file in the default directory. Then pop back out to the main Talon screen and select batch mode. You'll receive a confirmation window. When you select Yes, the Talon application slaves to auto mode. You can cancel auto mode from the main screen. You cannot, however, operate the telescope manually from the main screen while batch mode is in use.

The Talon program is rich with features, providing complete control over the operation of the telescope. Many of the finer features are outlined in the .pdf manual provided in the Talon .tgz file. It's worth a thorough read to understand the telescope interfaces to the control boards and to the software. The manual also contains in-depth information on image processing, solving for WCS solutions, automated and remote operations and finer calibration items that are beyond the scope of this article.

Talon represents a complete leap into the open-source world for astronomers, both professional and amateur. With the robust and networkable nature of Linux, Talon provides a stable platform from which we can do what we've been doing since the beginning of time—viewing, recording and discovering the heavens.

Tony Steidler-Dennison is the director of operations for Optical Mechanics, Inc., and the author of Lockergnome's *Penguin Shell* newsletter, a twice-weekly tome for Linux users. He maintains Talon on Sourceforge.net and the poli-tech blog "Frankly, I'd Rather Not" at **www.steidler.net**. He answers all e-mail sent to tony@steidler.net.

What's In The Stars For You This Year?
All Signs Point to Hurricane Electric!

Full 100 Mbps Port
With Full Cabinet
$2,500/month

Colocation Full Cabinet
Holds up to 42 1U servers
$400/month

Colocate Your Server
Up to 7U or 12 inches
256 kbps ~ 70GB
$200/month

Order Today!
email sales@he.net
or
call 510.580.4190

HE HURRICANE ELECTRIC INTERNET SERVICES

Monitoring Hard Disks with SMART

One of your hard disks might be trying to tell you it's not long for this world. Install software that lets you know when to replace it. **BY BRUCE ALLEN**

It's a given that all disks eventually die, and it's easy to see why. The platters in a modern disk drive rotate more than a hundred times per second, maintaining submicron tolerances between the disk heads and the magnetic media that store data. Often they run 24/7 in dusty, overheated environments, thrashing on heavily loaded or poorly managed machines. So, it's not surprising that experienced users are all too familiar with the symptoms of a dying disk. Strange things start happening. Inscrutable kernel error messages cover the console and then the system becomes unstable and locks up. Often, entire days are lost repeating recent work, re-installing the OS and trying to recover data. Even if you have a recent backup, sudden disk failure is a minor catastrophe.

Many users and system administrators don't know that Self-Monitoring, Analysis and Reporting Technology systems (SMART) are built in to most modern ATA and SCSI hard disks. SMART disk drives internally monitor their own health and performance. In many cases, the disk itself provides advance warning that something is wrong, helping to avoid the scenario described above. Most implementations of SMART also allow users to perform self-tests on the disk and to monitor a number of performance and reliability attributes.

By profession I am a physicist. My research group runs a large computing cluster with 300 nodes and 600 disk drives, on which more than 50TB of physics data are stored. I became interested in SMART several years ago when I realized it could help reduce downtime and keep our cluster operating more reliably. For about a year I have been maintaining an open-source package called smartmontools, a spin-off of the UCSC smartsuite package, for this purpose.

In this article, I explain how to use smartmontools' smartctl utility and smartd dæmon to monitor the health of a system's disks. See smartmontools.sourceforge.net for download and installation instructions and consult the WARNINGS file for a list of problem disks/controllers. Additional documentation can be found in the man pages (`man smartctl` and `man smartd`) and on the Web page.

Versions of smartmontools are available for Slackware, Debian, SuSE, Mandrake, Gentoo, Conectiva and other Linux distributions. Red Hat's existing products contain the UCSC smartsuite versions of smartctl and smartd, but the smartmontools versions will be included in upcoming releases.

To understand how smartmontools works, it's helpful to know the history of SMART. The original SMART spec (SFF-8035i) was written by a group of disk drive manufacturers. In Revision 2 (April 1996) disks keep an internal list of up to 30 Attributes corresponding to different measures of performance and reliability, such as read and seek error rates. Each Attribute has a one-byte normalized value ranging from 1 to 253 and a corresponding one-byte threshold. If one or more of the normalized Attribute values are less than or equal to its corresponding threshold, then either the disk is expected to fail in less than

Listing 1. Output of `smartctl -i /dev/hda`

```
Device Model:     IC35L120AVV207-0
Serial Number:    VNVD02G4G3R72G
Firmware Version: V240A63A
Device is:        In smartctl database [for details use: -P show]
ATA Version is:   6
ATA Standard is:  ATA/ATAPI-6 T13 1410D revision 3a
SMART support is: Available - device has SMART capability.
SMART support is: Enabled
```

Listing 2. Output of `smartctl -Hc /dev/hda`

```
SMART overall-health self-assessment test result: PASSED

General SMART Values:
Off-line data collection status: (0x82)  Offline data collection activity
                                         was completed without error.
                                         Auto Off-line Data Collection:
Enabled.
Self-test execution status:      (   0)  The previous self-test routine
                                         completed without error or no
                                         self-test has ever been run.
Total time to complete off-line
data collection:                (2855) seconds.
Offline data collection
capabilities:                    (0x1b) SMART execute Offline immediate.
                                        Automatic timer ON/OFF support.
                                        Suspend Offline collection upon new
                                        command.
                                        Offline surface scan supported.
                                        Self-test supported.
                                        No Conveyance Self-test supported.
                                        No Selective Self-test supported.
SMART capabilities:            (0x0003) Saves SMART data before entering
                                        power-saving mode.
                                        Supports SMART auto save timer.
Error logging capability:        (0x01) Error logging supported.
                                        General Purpose Logging supported.
Short self-test routine
recommended polling time:        (   1) minutes.
Extended self-test routine
recommended polling time:        (  48) minutes.
```

Listing 3. Output of `smartctl -A /dev/hda`

```
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x000b   100   100   060    Pre-fail  Always   -           0
  2 Throughput_Performance  0x0005   155   155   050    Pre-fail  Offline  -           225
  3 Spin_Up_Time            0x0007   097   097   024    Pre-fail  Always   -           293 (Average 270)
  4 Start_Stop_Count        0x0012   100   100   000    Old_age   Always   -           10
  5 Reallocated_Sector_Ct   0x0033   100   100   005    Pre-fail  Always   -           0
  7 Seek_Error_Rate         0x000b   100   100   067    Pre-fail  Always   -           0
  8 Seek_Time_Performance   0x0005   125   125   020    Pre-fail  Offline  -           36
  9 Power_On_Hours          0x0012   100   100   000    Old_age   Always   -           3548
 10 Spin_Retry_Count        0x0013   100   100   060    Pre-fail  Always   -           0
 12 Power_Cycle_Count       0x0032   100   100   000    Old_age   Always   -           10
192 Power-Off_Retract_Count 0x0032   100   100   050    Old_age   Always   -           158
193 Load_Cycle_Count        0x0012   100   100   050    Old_age   Always   -           158
194 Temperature_Celsius     0x0002   189   189   000    Old_age   Always   -           29 (Lifetime Min/Max 23/33)
196 Reallocated_Event_Count 0x0032   100   100   000    Old_age   Always   -           0
197 Current_Pending_Sector  0x0022   100   100   000    Old_age   Always   -           0
198 Offline_Uncorrectable   0x0008   100   100   000    Old_age   Offline  -           0
199 UDMA_CRC_Error_Count    0x000a   200   200   000    Old_age   Always   -           0
```

24 hours or it has exceeded its design or usage lifetime. Some of the Attribute values are updated as the disk operates. Others are updated only through off-line tests that temporarily slow down disk reads/writes and, thus, must be run with a special command. In late 1995, parts of SFF-8035i were merged into the ATA-3 standard.

Starting with the ATA-4 standard, the requirement that disks maintain an internal Attribute table was dropped. Instead, the disks simply return an OK or NOT OK response to an inquiry about their health. A negative response indicates the disk firmware has determined that the disk is likely to fail. The ATA-5 standard added an ATA error log and commands to run disk self-tests to the SMART command set.

To make use of these disk features, you need to know how to use smartmontools to examine the disk's Attributes (most disks are backward-compatible with SFF-8035i), query the disk's health status, run disk self-tests, examine the disk's self-test log (results of the last 21 self-tests) and examine the disk's ATA error log (details of the last five disk errors). Although this article focuses on ATA disks, additional documentation about SCSI devices can be found on the smartmontools Web page.

To begin, give the command `smartctl -a /dev/hda`, using the correct path to your disk, as root. If SMART is not enabled on the disk, you first must enable it with the `-s on` option. You then see output similar to the output shown in Listings 1–5.

The first part of the output (Listing 1) lists model/firmware information about the disk—this one is an IBM/Hitachi GXP-180 example. smartmontools has a database of disk types. If your disk is in the database, it may be able to interpret the raw Attribute values correctly.

The second part of the output (Listing 2) shows the results of the health status inquiry. This is the one-line Executive Summary Report of disk health; the disk shown here has passed. If your disk health status is FAILING, back up your data immediately. The remainder of this section of the output provides information about the disk's capabilities and the estimated time to perform short and long disk self-tests.

The third part of the output (Listing 3) lists the disk's table of up to 30 Attributes (from a maximum set of 255). Remember that Attributes are no longer part of the ATA standard, but most manufacturers still support them. Although SFF-8035i doesn't define the meaning or interpretation of Attributes, many have a de facto standard interpretation. For example, this disk's 13th Attribute (ID #194) tracks its internal temperature.

Studies have shown that lowering disk temperatures by as little as 5°C significantly reduces failure rates, though this is less of an issue for the latest generation of fluid-drive bearing drives. One of the simplest and least expensive steps you can take to ensure disk reliability is to add a cooling fan that blows cooling air directly onto or past the system's disks.

Each Attribute has a six-byte raw value (RAW_VALUE) and a one-byte normalized value (VALUE). In this case, the raw value stores three temperatures: the disk's temperature in Celsius (29), plus its lifetime minimum (23) and maximum (33) values. The format of the raw data is vendor-specific and not specified by any standard. To track disk reliability, the disk's firmware converts the raw value to a normalized value ranging from 1 to 253. If this normalized value is less than or equal to the threshold (THRESH), the Attribute is said to have failed, as indicated in the WHEN_FAILED column. The column is empty because none of these Attributes has failed. The lowest (WORST) normalized value also is shown; it is the smallest value attained since SMART was enabled on the disk. The TYPE of the Attribute indicates if Attribute failure means the device has reached the end of its design life (Old_age) or it's an impending disk failure (Pre-fail). For example, disk spin-up time (ID #3) is a prefailure Attribute. If this (or any other prefail Attribute) fails, disk failure is predicted in less than 24 hours.

The names/meanings of Attributes and the interpretation of their raw values is not specified by any standard. Different manufacturers sometimes use the same Attribute ID for different purposes. For this reason, the interpretation of specific Attributes can be modified using the `-v` option to `smartctl`; please see the man page for details. For example, some disks use Attribute 9 to store the power-on time of the disk in minutes; the `-v 9,minutes` option to `smartctl` correctly modifies the Attribute's interpretation. If your disk model is in the smartmontools database, these `-v` options are set automatically.

The next part of the `smartctl -a` output (Listing 4) is a log of the disk errors. This particular disk has been error-free, and

The SMART standard provides a mechanism for running disk self-tests and for monitoring aspects of disk performance.

the log is empty. Typically, one should worry only if disk errors start to appear in large numbers. An occasional transient error that does not recur usually is benign. The smartmontools Web page has a number of examples of `smartctl -a` output showing some illustrative error log entries. They are timestamped with the disk's power-on lifetime in hours when the error occurred, and the individual ATA commands leading up to the error are timestamped with the time in milliseconds after the disk was powered on. This shows whether the errors are recent or old.

The final part of the `smartctl` output (Listing 5) is a report of the self-tests run on the disk. These show two types of self-tests, short and long. (ATA-6/7 disks also may have conveyance and selective self-tests.) These can be run with the commands `smartctl -t short /dev/hda` and `smartctl -t long /dev/hda` and do not corrupt data on the disk. Typically, short tests take only a minute or two to complete, and long tests take about an hour. These self-tests do not interfere with the normal functioning of the disk, so the commands may be used for mounted disks on a running system. On our computing cluster nodes, a long self-test is run with a cron job early every Sunday morning. The entries in Listing 5 all are self-tests that completed without errors; the LifeTime column shows the power-on age of the disk when the self-test was run. If a self-test finds an error, the Logical Block Address (LBA) shows where the error occurred on the disk. The Remaining column shows the percentage of the self-test remaining when the error was found. If you suspect that something is wrong with a disk, I strongly recommend running a long self-test to look for problems.

The `smartctl -t offline` command can be used to carry out off-line tests. These off-line tests do not make entries in the self-test log. They date back to the SFF-8035i standard, and update values of the Attributes that are not updated automatically under normal disk operation (see the UPDATED column in Listing 3). Some disks support automatic off-line testing, enabled by `smartctl -o on`, which automatically runs an off-line test every few hours.

The SMART standard provides a mechanism for running disk self-tests and for monitoring aspects of disk performance. Its main shortcoming is that it doesn't provide a direct mechanism for informing the OS or user if problems are found. In fact, because disk SMART status frequently is not monitored, many disk problems go undetected until they lead to catastrophic failure. Of course, you can monitor disks on a regular basis using the smartctl utility, as I've described, but this is a nuisance.

The remaining part of the smartmontools package is the smartd dæmon that does regular monitoring for you. It monitors the disk's SMART data for signs of problems. It can be configured to send e-mail to users or system administrators or to run arbitrary scripts if problems are detected. By default, when smartd is started, it registers the system's disks. It then checks their status every 30 minutes for failing Attributes, failing health status or increased numbers of ATA errors or failed self-tests and logs this information with SYSLOG in /var/log/messages by default.

You can control and fine-tune the behavior of smartd using the configuration file /etc/smartd.conf. This file is read when smartd starts up, before it forks into the background. Each line contains Directives pertaining to a different disk. The configuration file on our computing cluster nodes look like this:

```
# /etc/smartd.conf config file
/dev/hda -S on -o on -a -I 194 -m sense@phys.uwm.edu
/dev/hdc -S on -o on -a -I 194 -m sense@phys.uwm.edu
```

The first column indicates the device to be monitored. The `-o on` Directive enables the automatic off-line testing, and the `-S on` Directive enables automatic Attribute autosave. The `-m` Directive is followed by an e-mail address to which warning messages are sent, and the `-a` Directive instructs smartd to monitor all SMART features of the disk. In this configuration, smartd logs changes in all normalized attribute values. The `-I 194` Directive means ignore changes in Attribute #194, because disk temperatures change often, and it's annoying to have such changes logged on a regular basis.

Normally smartd is started by the normal UNIX init mechanism. For example, on Red Hat distributions, `/etc/rc.d/init.d/smartd start` and `/etc/rc.d/init.d/smartd stop` can be used to start and stop the dæmon.

Further information about the smartd and its config file can be found in the man page (`man smartd`), and summaries can be found with the commands `smartd -D` and `smartd -h`. For example, the `-M test` Directive sends a test e-mail warning message to confirm that warning e-mail messages are delivered correctly. Other Directives provide additional flexibility, such as monitoring changes in raw Attribute values.

What should you do if a disk shows signs of problems? What if a disk self-test fails or the disk's SMART health status fails? Start by getting your data off the disk and on to another system as soon

Listing 4. Output of `smartctl -l error /dev/hda`

```
SMART Error Log Version: 1
No Errors Logged
```

Listing 5. Output of `smartctl -l selftest /dev/hda`

```
SMART Self-test log, version number 1
Num  Test_Description    Status       Remaining  LifeTime(hours)  LBA_of_first_error
# 1  Extended off-line   Completed    00%        3525             -
# 2  Extended off-line   Completed    00%        3357             -
# 3  Short off-line      Completed    00%        3059             -
```

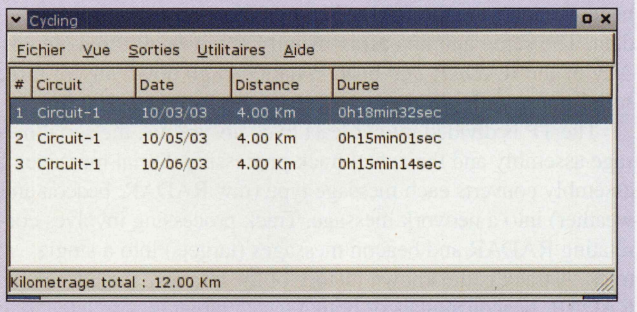
as possible. Second, run some extended disk self-tests and see if the problem is repeatable at the same LBA. If so, something probably is wrong with the disk. If the disk has failing SMART health status and is under warranty, the vendor usually will replace it. If the disk is failing its self-tests, many manufacturers provide specialized disk health programs, for example, Maxtor's PowerMax or IBM's Drive Fitness Test. Sometimes these programs actually can repair a disk by remapping bad sectors. Often, they report a special error code that can be used to get a replacement disk.

This article has covered the basics of smartmontools. To learn more, read the man pages and Web page, and then write to the support mailing list if you need further help. Remember, smartmontools is no substitute for backing up your data. SMART cannot and does not predict all disk failures, but it often provides clues that something is amiss and has helped to keep many computing clusters operating reliably.

Several developers are porting smartmontools to FreeBSD, Darwin and Solaris, and we recently have added extensions to allow smartmontools to monitor and control the ATA disks behind 3ware RAID controllers. If you would like to contribute to the development of smartmontools, write to the support mailing list. We especially are interested in information about the interpretation and meaning of vendor-specific SMART Attribute and raw values.

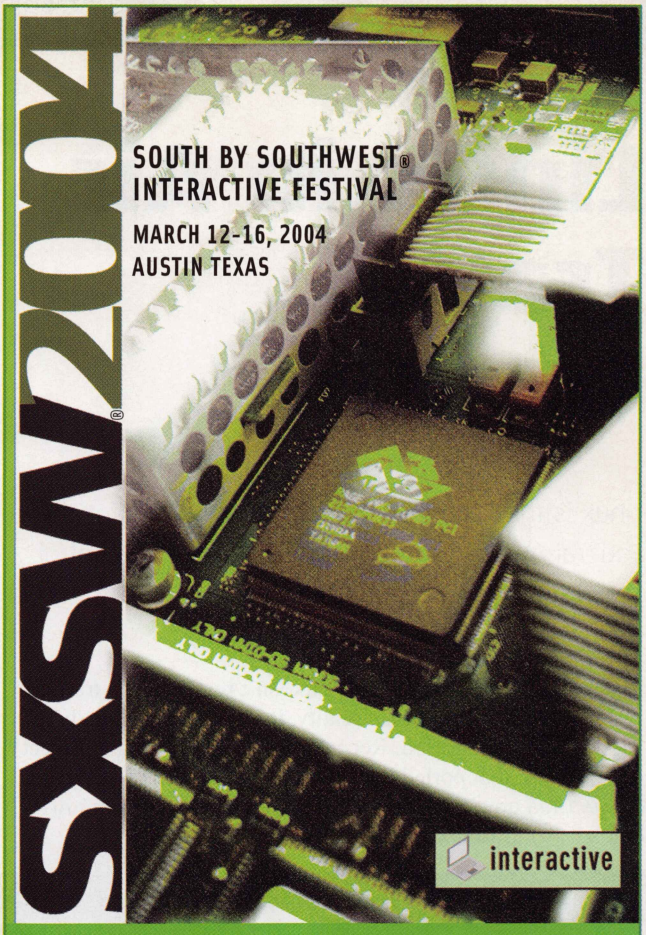
Bruce Allen is a professor of Physics at the University of Wisconsin - Milwaukee. He does research work on gravitational waves and the very early universe, and he has built several large Linux clusters for data analysis use.

Cycling: juvenis.dyndns.org/index.php?dirid=1&artid=3



Cycling is designed to help you with your cycling training, by keeping track of courses run, maximum speeds, distances and more. In fact, although designed for cycling, there's no reason it can't be put to use for running, swimming or almost any sport in which you might want to record time, distance and speed. Requires: libgtk-x11-2.0, libatk-1.0, libgdk-x11-2.0, libgdk_pixbuf-2.0, libm, libpangoxft-1.0, libpangox-1.0, libpango-1.0, libgobject-2.0, libgmodule-2.0, libdl, libglib-2.0, glibc, libXi, libXext, libXft, libXrender, libfontconfig, libX11, libfreetype, libz and libexpat.

—DAVID A. BANDEL



SXSW.COM

11th ANNUAL
SXSW INTERACTIVE FESTIVAL

CREATIVITY IS KING AT THE SOUTH BY SOUTHWEST (SXSW) INTERACTIVE FESTIVAL. Come to SXSW Interactive and be inspired by the people who are doing the most innovative work in web-based content and design.

2004 KEYNOTE SPEAKERS
Jonathan Abrams (friendster.com)
Zach Exley & Eli Pariser (moveon.org)
Brenda Laurel (author)
Howard Rheingold (author)

Other confirmed speakers include Cory Doctorow, Rael Dornfest, Dan Gillmor, Joi Ito, Neal Pollack, Virginia Postrel, Bruce Sterling, Jeff Veen, and Jeffrey Zeldman.

For more info, see www.sxsw.com/interactive.

THE AUSTIN CHRONICLE austinchronicle.com

Miller Lite

Linux in Air Traffic Control

Linux is the natural development platform for the Federal Aviation Administration's air traffic control software. **BY TOM BRUSEHAVER**

Many people say Linux isn't ready for air traffic control, but in reality, it is ready and already is being used. Recently I was involved in a project that ported the FAA's Common ARTS software (www1.faa.gov/ats/atb/Sectors/Automation/CommonArts/index.htm) to Linux.

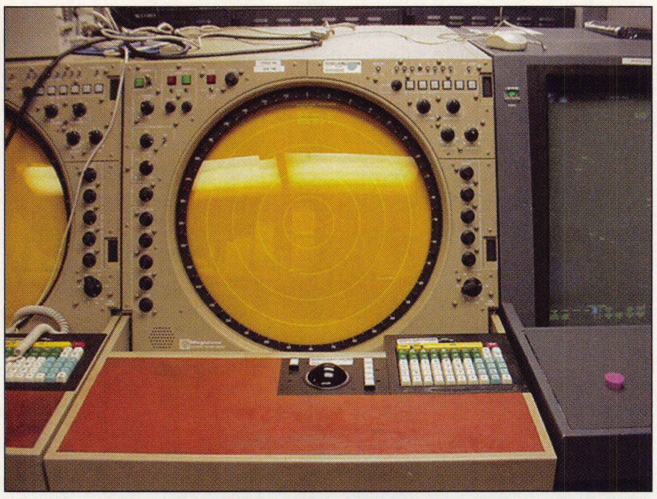
A Controller in Atlanta

When most of us think of air traffic control systems, we usually see round screens with a sweep. Some people know what the RADAR antenna array looks like. Between the RADAR antenna and the display are a few computers to make things easier. The Digital Signal Processors (DSPs) are in the antenna building, and automation systems are used as well. This article describes how the automation system works, and where Linux was used.

History

The Automated Radar Terminal System (ARTS) got its start back in 1964 on Univac computers. The system eventually went nationwide in 1973. The original computers have been upgraded and continue to be used today. Many of the larger sites moved from the Univac computers to microprocessors during the 1980s. All of the legacy ARTS software was ported or rewritten in C to run under the real-time LynxOS during the move to microprocessors.

The move to LynxOS was fortuitous, as it provided a POSIX base to which further porting could be done. Using



Full digital ARTS display. The ACDs (right side) are replacing these vector displays. All the knobs are now menus on the display.

LynxOS also allowed developers to choose which microprocessors to use. Initially, the software ran on Motorola 68K CPUs, and it currently runs on PowerPCs.

Architecture

The Common ARTS system is a highly distributed, networked, multithreaded, real-time system. Absolute reliability is a requirement. Dual networks are used, and under normal conditions, two backups are assigned to each specific task. The software is designed such that some functions of one subsystem can be taken over by another subsystem.

The RADAR data comes in over serial lines to each of the track processors (TPs). Normally, four serial lines go from each RADAR to each TP. The data on the RADAR include the raw RADAR signal, a transponder beacon (each aircraft should have a transponder that returns an ID and an altitude in the beacon each time the RADAR pings the aircraft) and weather data. The same antenna array usually receives all three signals. Any of the RADAR and beacon data can go down any of three serial lines, with weather transferring on its own serial line.

The TP is divided into at least two subsystems, the serial message assembly and the actual track processing. Serial message assembly converts each message type (raw RADAR, beacon and weather) into a network message. Track processing involves correlating RADAR and beacon messages (targets) into a single track. A track is the known history of the aircraft. Targets can be RADAR, beacon signals or both. Once a target is correlated to a track, another track message is put on the network.

Messages usually are broadcast on the network. They are sent using UDP on both trunks, and each message has a unique packet ID. Each computer on the network listens to both trunks and keeps a record of the packet IDs it has heard from each unique network ID. If there is a gap in packet IDs, a chassis may request a rebroadcast. Duplicate packet IDs are ignored, assuming it was the message on the other trunk or some other chassis requesting a rebroadcast.

Each system must broadcast heartbeats on the network. If a heartbeat is missed, it is assumed the system is down, and one of the other systems sends a message to have a standby system take over.

The next processor to deal with the network messages is the common processor (CP). The CP does many things, such as matching flight plans to tracks, sending conflict alerts (CAs), minimum safe altitude warning (MSAW) and monitoring some of the Common ARTS system heartbeats. The biggest thing the CP does is determine airspeed and direction of a track.

CA uses the speed and direction to look at other tracks to determine the possibility of a conflict. If a conflict is detected, the CP broadcasts a CA message indicating the aircraft in conflict. When aircraft are traveling at 300KTS—about five miles a minute—it is important to be looking a minute or two ahead.

MSAW uses a site-adapted map to learn terrain in the area. The terrain can be hills and mountains as well as towers and buildings. For transponder-equipped aircraft, the MSAW system looks at the altitude and the position and determines if the aircraft may be too low. If an aircraft is determined to be too low, an MSAW message is broadcast.

The last major system in the back room is the system monitor and control (SMC). The main purpose of the SMC is to monitor and control the other systems. It is a gateway to the SMC display PC, a GUI for monitoring the network and its current state. Current state indicates the systems that are on-line, off-line, standby or idle. If a heartbeat is missed, the SMC instructs a standby system to take over. A system operator can instruct a manual switch at anytime, load new software or reboot systems from this PC. The SMC also is used for recording all the data that crosses the network.

What most of us think of when we think of air traffic control are the displays: a room full of round vector displays and guys in white shirts watching them. More and more sites are using large 20"-square color displays. The new color displays are 2048×2048 pixel, X window displays. The design of the display processing software (DPS) is such that parts of it can be used by any display type, the color square display, like an ARTS Color Display (ACD) or the round vector display, like a Full Digital ARTS Display (FDAD).

The DPS receives the broadcast messages and displays appropriate images depending on the state of the system. In normal operation, the display includes a track indication, a direction history indication, a full or partial datablock and the state of the various systems.

When aircraft are traveling at 300KTS—about five miles a minute—it is important to be looking a minute or two ahead.

The flight plan information message from the CP is displayed in the full datablock near the aircraft for which the TP created the track. The broadcast SMC overall state can be displayed as well. The network can use one or several hundred displays.

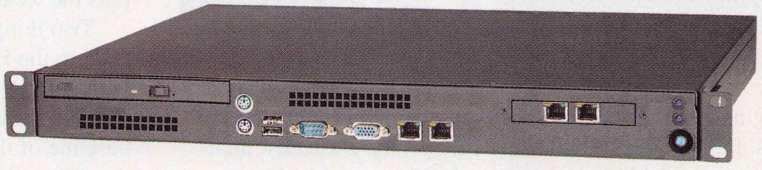
In addition, each system can be run on one or more CPUs. If the CPU is powerful enough, all the systems and subsystems can be run on a single CPU.

Porting

Initially, the reason for the port to Linux was to allow developers to test and debug the systems at their desks before testing on the target hardware. The target hardware is Motorola OEM boards in a VME chassis running LynxOS. The systems are relatively expensive, so neither the FAA nor Lockheed-Martin wants to have a bunch sitting around. Instead, several test systems are used almost full-time for integration testing and development.

Because the IS department gives developers a Microsoft Windows NT PC, an attempt was made to port the software to NT. Most of the port was completed when I started working for the company. For testing some things, NT worked fine. An adapter layer was used to make the POSIX threads, file I/O and graphics behave like the target system, so it wasn't good at testing those features.

When I started contracting at Lockheed-Martin, I was placed in the

Ultra Dense, Powerful, Reliable...
Datacenter Management Simplified!
15" Deep, 2-Xeon/Opteron or P4 (w/RAID) options

Customized Solutions for...
Linux, BSD, W2K
High Performance Networking Solutions
• Data Center Management
• Application Clustering
• Network and Storage Engines
Rackmount Server Products
• 1U Starting at $499: C3-1GHz, LAN, 256MB, 20GB IDE
• 2U with 16 Blades, Fast Deployment & more...
iron SYSTEMS™
Iron Systems, Inc.
2330 Kruse Drive, San Jose, CA
www.ironsystems.com
CALL: 1-800-921-IRON

Actual screenshot of the 8-bit VGA Linux DPS software. Compared to the ACD display, only the all-brown weather is different. The blue history trails look similar, and the green beacon and RADAR images are the same. The menu is transparent to the map, but not weather.

A real controller watching an ACD.

messaging layer group, the group that maintains the communications, threading and file I/O for the system. Basically, none of my testing could be done on the desktop, and I had to use the target hardware. Initially, a side project was working to see if it was possible, and I was given one old (200MHz Pentium) PC for development purposes.

Most of the code compiled just fine, although there were some issues with POSIX standards. LynxOS 2.4 and 3.0 used an older standard, whereas Linux uses the current one. I initially was doing the development on Red Hat 7.0 with a 2.2 kernel, and it didn't support named semaphores or named shared memory segments. In a distributed system like ours, it is easier to use standard names within a processor than some other communication mechanism to know where the shared memory and semaphores are. I did cobble together a named shared memory compatibility layer, and I found a Russian site with a named semaphore compatibility layer.

During development I moved to Red Hat 7.1, which was supposed to support named shared memory, but there was an incompatibility between glibc and the header files. I was able to look in the source to find this problem, and I posted a note to the kernel mailing list. Someone beat me to it, though. To keep things stock, allowing anyone to pick up any Red Hat version without this fix, I left my cobbled-together version in the code.

The target hardware was all big endian (Motorola 68K and PPC) and the Linux PC was an x86 little endian, so I needed to do some byte swapping to make the whole system work. Many of the files are stored in binary (maps, adaptation, and so on) format. The networking layer already had a byte-swapping mechanism built in, and it worked great.

Once I had all the messaging code compiled and running, I needed an application. The FAA agreed to fund the further development of the TP, CP, SMC and DPS systems for desktop testing and debugging. The systems all ported well, but the DPS had some issues with X displaying. Normally the large 2048×2048 pixel display is run on special hardware with two or three pseudo-color physical planes. If the maps and menus are drawn on the bottom plane, the weather on another plane and the aircraft on the top plane, the whole display doesn't need to be redrawn if an aircraft moves. To make this plane idea work, the color map was split into three parts. Being pseudo-color (8-bit) limited the number of colors in each plane. The map and menu plane got one color (white), the weather got another color (brown) and the primary display plane got 78 colors.

So, we needed to have more color table adjustments, as normally the primary display used 250 colors. The large display has an animated fading history trail that emulates the fading phosphors on the vector display. It takes 128 color cells to make the animation work. For this application I made one cell and no animation. It looks amazingly good. Between finding all the reds that were similar, greens that were similar, and yellows and blues and whites and grays, I trimmed the table down to the 78 available.

Once all this was done, I was given a two-plane video card to see if the system would still work. A change to one compile-time flag had the whole thing working. The two-plane card puts the weather maps and menus on the same plane.

Two things happened about this time. I delivered the Linux code to the FAA, and a couple other developers were tasked with getting Linux working on the target PowerPC hardware. The FAA found a few updates I hadn't kept up with in the baseline of the code, and we were able to work together on that. The other developers found most of my #ifdefs were specific to Linux and not the machine architecture. Therefore, I was able to hold back my changes to the FAA and make the proper #ifdefs when the FAA finally took it.

The PowerPC Linux Project was an attempt to improve the data recording tasks that the SMC handles. The current system uses consumer-grade, off-the-shelf optical disks that aren't suitable for 24/7 writing. The new system incorporates SAN disks, which are more suitable to air traffic control needs. Although technically a success, the project is on hold for now.

In the spring of 2003, the FAA began using the Common ARTS on Linux for a noncritical subsystem, an inexpensive gateway system feeding ARTS data to other systems. Full certification may happen eventually.

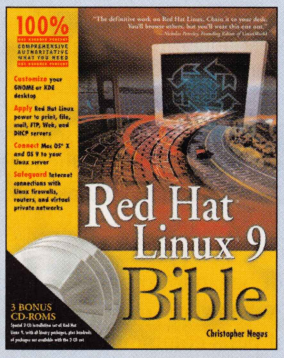
Tom Brusehaver is a coder grunt who has been writing code since before the PC. He mostly does contract work these days, preferring embedded systems. He is married and has grown kids, two cats and a dog. He is building an airplane when the weather is nice.

BOOK REVIEW

Red Hat Linux 9 Bible

by Christopher Negus

John Wiley & Sons, 2003
ISBN: 0-7645-3938-8
$49.99 US



At 1,000+ pages, calling this book a bible is not an exaggeration. Browsing its pages, I quickly realized that it is broken into three sections, the first oriented toward users, the second toward system administrators and the third toward networks. The book comes with three CDs for installing Red Hat Linux 9.

The first thing I look for in a book such as this is the section that explains what is new in Red Hat 9. The book explains it all on page 12, "Although you don't see it, the Native POSIX Thread Library is the most significant addition to Red Hat Linux 9 (and the main reason why the release is called 9 instead of 8.1)." Thank you. The book also details the components of Red Hat 9 that have been upgraded, dropped or whose days are numbered.

Chapter 2 is worth reading in-depth, especially for a novice to Linux, for grounding in the concepts before proceeding with the tasks. Chapters 3–9 cover the desktop, basic commands and how to run applications. These chapters help users who are a bit disoriented when faced with a new computing environment.

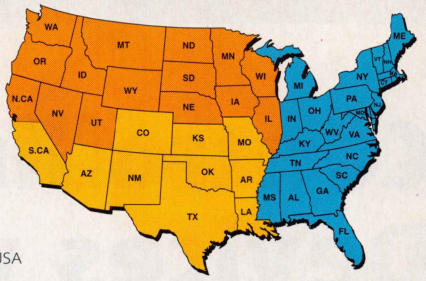
The important chapters in this book—dealing with system and network administration—are missing what most books on Linux system administration seem to miss: information about troubleshooting and gathering system information.

Chapter 13, "Backing Up and Restoring Files", is comprehensive, and I especially enjoyed the discussion of mirrordir. The best thing about this chapter is many solutions are discussed, and the reader can select from several options. I also liked the quick reference charts for pax, dump and Amanda. This is not the only chapter with these charts; they are used liberally throughout the book and are quite handy.

Network services are easily a topic for a whole other book, and they consume more than a third of this book. Chapter 15 discusses setting up a LAN and includes information on wireless networks, which to me makes it timely and useful. I'd call this a handy reference to consider and keep available, especially for those new to Linux.

—FRANK CONLEY

LINUX JOURNAL



PO Box 55549
Seattle, WA 98155-0549 USA
www.linuxjournal.com

ADVERTISING SERVICES

VP OF SALES AND MARKETING
Carlie Fairchild, carlie@ssc.com
+1 206-297-8655,
+1 206-782-7191 FAX

FOR GENERAL AD INQUIRIES
e-mail ads@ssc.com
or see www.linuxjournal.com/advertising

To receive further information from any of our advertisers, please visit our Reader Service page to access this service: **www.linuxjournal.com/advertising/rs**

REGIONAL ADVERTISING SALES

NORTHERN USA
Joseph Krack, joseph@ssc.com
866-423-7722 (toll-free),
866-423-7722 FAX

SOUTHERN USA
Annie Tiemann, annie@ssc.com
866-965-6646 (toll-free),
866-422-2027 FAX

EASTERN USA AND CANADA
Martin Seto, mseto@ssc.com
+1 905-947-8846,
+1 905-947-8849 FAX

Please direct international advertising inquiries to VP of Sales and Marketing, Carlie Fairchild.

Advertiser	Page #
ASA Computers www.asacomputers.com	53
Angstrom Microsystems www.angstrom.com	1
Appro International, Inc. appro.com	49, 83
Arkeia Corporation www.arkeia.com/demo	25
Aspen Systems, Inc. www.aspsys.com	7
Central Command, Inc. www.centralcommand.com	63
Ciara Technology www.ciara-tech.com	61
Computer Associates International www.ca.com/linux	35
Consensys Computers, Inc. www.consensys.com	C3
Cyclades Corporation www.cyclades.com	2, 9
EmperorLinux www.emperorlinux.com	15
Etnus www.etnus.com	67
FairCom Corporation www.faircom.com	29
Free Software Foundation www.gnupress.org	86
Hentzenwerke www.hentzenwerke.com	17
Hurricane Electric www.he.net	73
ImageStream Internet Solutions www.imagestream-is.com	39
Integrated Computer Solutions, Inc. www.ics.com	51
Interland www.interland.com	47
Iron Systems, Inc. www.ironsystems.com	79
Lantech PC Systems www.lantechpc.com	27
Layer 42 Networks www.layer42.net	59
LinuxWorld Conference & Expo www.linuxworldexpo.com	32, 33
LinuxCertified, Inc. www.linuxcertified.com	93
Microway, Inc. www.microway.com	C4
Mikro Tik www.routerboard.com	71
Monarch Computers www.monarchcomputer.com	22, 23
O'Reilly & Associates conference.oreilly.com/oscon	89
Paracel www.paracel.com	5
Parasoft Corporation www.parasoft.com/linux7	55
The Portland Group www.pgroup.com	43
RackSaver www.racksaver.com	C2, 45
Rackable Systems www.rackable.com	19
Rackspace Managed Hosting www.rackspace.com	13
Rae Internet, Inc www.raeinternet.com	85
SBE, Inc. www.sbei.com	11, 21
SXSW Festivals and Conferences www.sxsw.com	77
Software Development Conference and Expo West www.sdexpo.com	57
SuSE www.suse.com	41
SWsoft www.sw-soft.com	37
Technologic Systems www.embeddedx86.com	31

Controlling Devices with Relays

Ring bells and more with a simple C program and inexpensive hardware. BY JASON ELLISON

I recently was given the curious task of making noise on a factory floor at set intervals. The company wanted to use this noise to provide important auditory cues for specific events. These events included the time employees should clock in, clock out, take a break and return from break. The solution I implemented has been working well, and everyone involved has been pleased with the results. The gratitude I receive from employees that clock in and clock out is related directly to the inaccuracy of their own personal timepiece—those with fast watches appreciate the apparent extra time they have before clocking in. Later, though, they curse me for being delayed in clocking out when the watches on their wrists clearly show it is time to go.

The company I work for uses barcode clock terminals placed throughout the plant to record the time that employees clock in and clock out. In addition, the barcode clocks also record what job each individual employee is performing. When an employee clocks in or out, the employee scans his or her identification badge and a barcode representing the job the person is doing. The clock terminal copies this data to a buffer, along with a timestamp of when the items were scanned. CMINet, a data collection system running on two Microsoft Windows NT systems, is used to acquire the data from the clocks. The clocks are polled every minute or so to retrieve the data held in the clock terminal's buffer. During polling, the clocks also are set to the local time of the machine polling the clocks. The data acquired from the clocks is later imported into the enterprise resource planning (ERP) system for job costing and payroll.

The company's policy on tardiness states, "employees will be considered tardy if they clock in one minute or more after their assigned start time." Due to the company policy concerning tardiness, the event bells needed to be synchronized perfectly with the barcode clocks and the machines running CMINet. This synchronization would, in theory, alleviate the discrepancies between the recorded clock-in time, recorded clock-out times and the true local time. Correct time synchronization is key, but it is not discussed thoroughly in this article.

Considering the one-minute-or-more-late-equals-tardy rule and the fact that employees invariably become time-challenged when returning from break, it was decided that scheduled bell soundings would be useful. Therefore, a mission-critical noise-making system was needed, and I was given the task of implementing it.

Implementation Concepts

Based on the issues at hand, this project had to meet the following requirements:

1. Provide auditory cues using a bell system. The bells need to be audible and annoying enough to be noticed even after prolonged exposure.

2. Provide a simple way of closing an electrical circuit over a computer.

3. Provide accurate time synchronization for all time-sensitive systems, including the machine that would trigger the bells, the machines running CMINet and, indirectly, the barcode clocks.

With the requirements defined I began to formulate a plan. Five new 12VAC bells and a 12VAC power supply were purchased. These are the same type of bells commonly used in alarm systems or any sounding system. The bells physically hammer themselves in the same manner as an old alarm clock, and they pump out the decibels. The bells needed to be wired so that connecting two wires would ring all the bells in the plant.

To control the bells, we needed to close the bells' circuit temporarily. I was pointed toward a specific relay device, the AR-2, that can be controlled by a computer over a serial port. This device would be used to complete the bell circuit, causing them to ring. The AR-2 is sold by Electronic Energy Control, Inc. (EECI) for about $44 US. It has two relays on a circuit board, and both can be connected as normally open (NO) or normally closed (NC) relays. Only one relay wired as NO would be used for this project. An enclosure (EN-B) to house the board also was purchased. An emergency on/off switch was mounted on the enclosure for use in case something went terribly wrong. The AR-2 is documented well enough that I knew it would be possible to write a program in C to control it. This program could be scheduled using cron jobs.

For time synchronization, the NTP protocol was chosen. NTP clients and servers are unlikely to have interoperability problems due to a well-defined protocol that is widely adopted. In addition, NTP clients can handle network latency, daylight saving time and local time zones. Free clients are available for almost every major operating system in use today. Computers using these clients can readily obtain time sources accurate to at least the millisecond. Good accuracy is exactly what we need when trying to synchronize multiple systems.

Implementation Details

The five 12VAC bells were installed throughout the plant and in the designated break areas so that everyone would hear them. The bells were wired in mixed series and parallel with a 12VAC power supply supplied by the vendor. One lead from the power supply was left unconnected to provide a point at which the bells could be turned on and off.

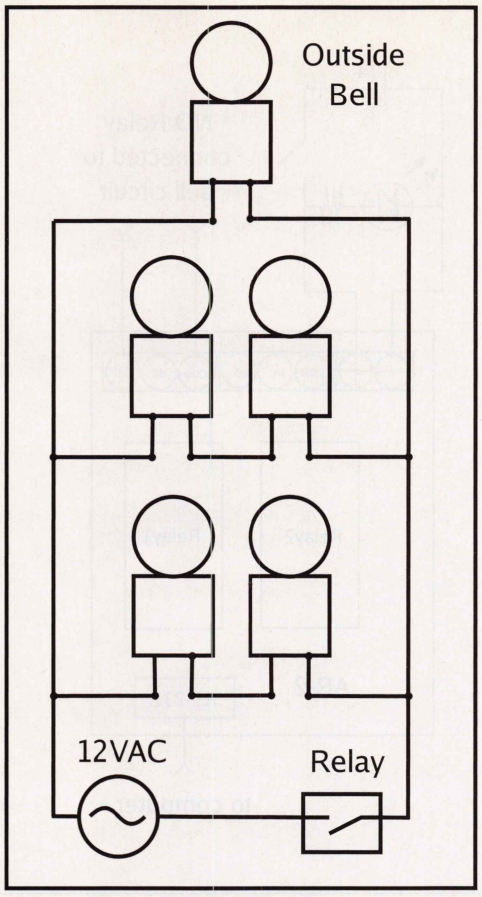


Figure 1. Schematic of the Bell System

I obtained one of our company's spare Dell PCs to use for this project, a basic PC with a Pentium III 750MHz processor, 128MB of SDRAM, 40GB IDE hard drive and an IDE CD-ROM. I used what was readily available, and really, the computer is a lot more powerful than necessary.

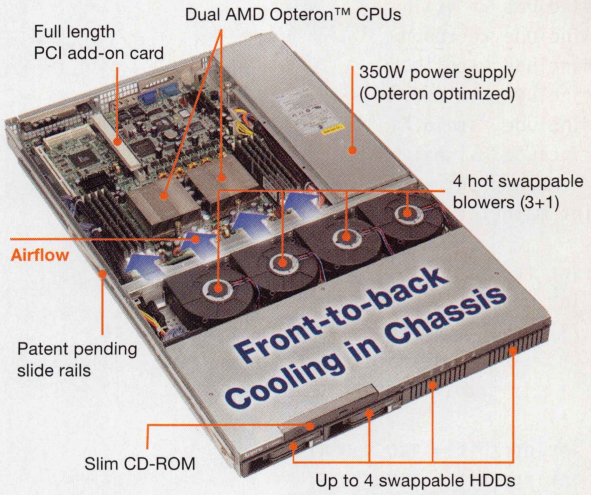
The AR-2 relay device attaches to the computer's serial port and is controlled by the DTR (data terminal ready) and RTS (ready to send) control lines. Setting RTS to high energizes the first relay and setting DTR to high energizes the second relay. Closing the bell circuit requires only one relay, so I needed to be able to toggle only the RTS signal. Example programs in GWBASIC and Turbo C were provided with the documentation. I wanted to write a program using GCC to control the relay but had little experience in C programming, so I searched the Web for some well-documented examples of C programs that made use of a serial port.

The first interesting program I found was upscheck.c from Harvey J. Stein's UPS HOWTO. upscheck, itself a modified version of Miquel van Smoorenburg's powerd.c, makes use of both the RTS signal and the DTR signal. The program originally was used to diagnose or examine UPS communications with a PC over a serial port. When starting the program, you indicate with parameters to which serial device the UPS is attached and whether to set RTS and/or DTR high. After the program opens the port and sets or clears the indicated serial control lines, the program would monitor another control line for feedback from the UPS. The program did everything I needed and more. All I had to do was remove the monitoring part, and I had my C

Appro
High-Performance Servers and Cluster Solutions

32-bit / 64-bit AMD Opteron™ Based Servers
Why Pay More for Less Cooling?

Introducing the New Rack Optimized Servers
Appro 1100H – Cluster Engine Servers
Appro 1122H – Enterprise Class Servers



- Up to two dual AMD Opteron™ processors
- Up to 16GB PC2700 ECC, DDR RAM
- Up to 4 HDDs - IDE or SCSI
- Dual Gigabit ports
- 24x slim CD-ROM optional
- Integrated Cluster Solutions are available

Appro's new design distributes cooling air evenly over the two CPU heat sinks. It assures performance reliability, flexibility and headroom necessary to keep high-end server customers up and running without worrying about thermal issues.

Available early January/04
Pre-orders are available now!

AMD Opteron™ Processors
Provides maximum flexibility and run their 32-bit applications faster while managing the transition cost of 64-bit migration.

AMD 64 Opteron

APPRO
HPC Cluster Solutions

For more information, visit **www.appro.com** or call Appro Sales at **800.927.5464**.

program to control both relays on the AR-2 (Listing 1). The program is compiled with the following command:

```
[root@pluto ar-2]# gcc -o ar-2 ar-2.c

[root@pluto ar-2]# ./ar-2
Usage: ar-2 <device> <bits-to-set> <hold-time>
```

Listing 1. ar-2.c

```
#include <sys/types.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <errno.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <signal.h>
/* Main program. */
int main(int argc, char **argv)
{
 int fd;
 int sleep_time;
 /* These TIOCM_* parameters are defined in
  *   <linux/termios.h>, which
  *  is indirectly included here.
  */
 int dtr_bit = TIOCM_DTR;
 int rts_bit = TIOCM_RTS;
 int set_bits;
 int flags;
 int status, oldstat = -1;
 int count = 0;
 int pc;
 if (argc < 3) {
  fprintf(stderr, "Usage: ar-2 <device> "
                  "<bits-to-set> <hold-time>\n");
  exit(1);
 }
 /* Open monitor device. */
 if ((fd = open(argv[1], O_RDWR | O_NDELAY)) < 0) {
  fprintf(stderr, "ar-2: %s: %s\n",
          argv[1], sys_errlist[errno]);
  exit(1);
 }
 /* Get the bits to set from the command line. */
 sscanf(argv[2], "%d", &set_bits);
 /* get delay time from command line. */
 sscanf(argv[3], "%d", &sleep_time);
 ioctl(fd, TIOCMSET, &set_bits);
 sleep(sleep_time);
 close(fd);
}
```

The program compiled with no errors, and better yet, it ran with no errors. Using the program is pretty straightforward, but let's briefly discuss it. AR-2 requires three parameters: device, bits-to-set and hold-time. Device refers to the special character device that addresses the serial device to which the AR-2 is attached. For example, if the device is attached to COM1, the device would be /dev/ttyS0. Bits-to-set is a little more complicated. It is a decimal number that must be converted to binary to be understood. Bit 2 controls RTS or relay one, and bit 3 controls DTR or relay two. So the decimal number 4, which in binary is 100, has the first bit set to zero, the second bit set to zero and the third bit set to one. Therefore, if the bit-to-set parameter is the decimal number 4, the DTR line would be placed high and the RTS line would remain low. The last parameter, Hold-time, simply is the number of seconds the modem control lines should be held in the state requested by the bits-to-set parameter.

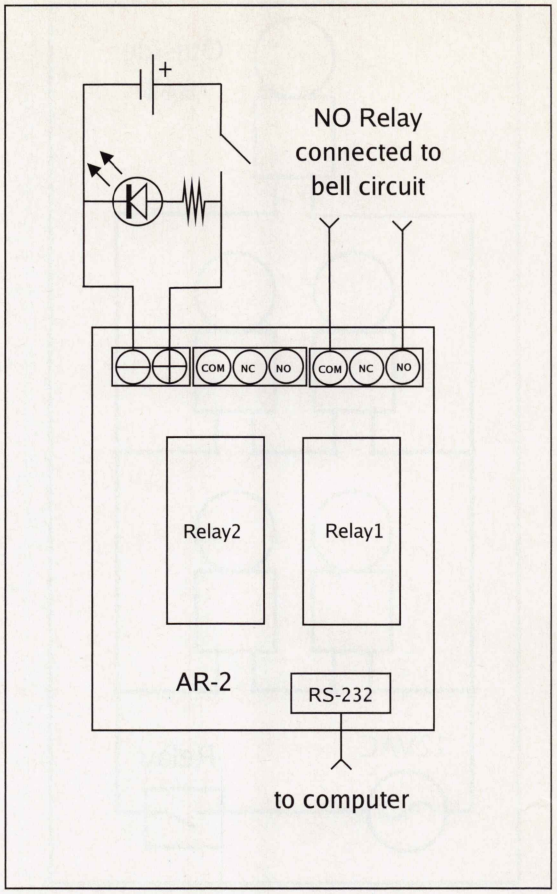


Figure 2. Connections to the AR-2 Board

I applied power to the AR-2 board and connected it to the serial port to test that the program worked as I had hoped. With the AR-2 connected to ttyS0 (COM1), I attempted to energize the first relay by placing RTS high by setting bit 2 with the command `./ar-2 /dev/ttyS0 2 5`. The program successfully placed RTS high for five seconds. After the program exited, it placed the RTS line in its original low state, de-energizing the relay. Next, I placed DTR high by setting bit 4 using the command `./ar-2 /dev/ttyS0 4 5`. Both RTS and DTR could be placed high by setting bits 4 and 2 at the same time, using the decimal number 6 for the bits-to-set parameter. The LEDs on the board are helpful in determining that both the AR-2 program and AR-2 board are operational.

Thanks to upscheck.c and its authors, I was able to place both the RTS and DTR control lines in a low or high state for a specified amount time. Without the source code listed

in the UPS HOWTO, things would not have gone so smoothly. With a little effort, I was able to find C source code that was close to what I needed and modify it to meet my needs. I now can control two relays from a single serial port.

At this point, the bells could be attached to the first relay on the AR-2 using the NO contacts. Before connecting it all, though, I wanted to put the AR-2 in an enclosure, a nice little box to hold all this stuff so the electronics aren't exposed. I used the EN-B enclosure recommended and sold by EECI. It consists of two plastic pieces, a top and bottom, with black plastic inserts to cover openings on two sides. The enclosure is rather large for this job, but it works. I had to cut the black plastic inserts so that the external connectors would be accessible. Also, a hole was cut for the serial connector on the AR-2 that would be used for the connection to the computer's serial port. A hole was drilled for the female power connector that powers the AR-2 board. To connect the relay and the bell circuit, I chose a red and black banana clip, which can be found at most electronics stores. Two more holes were drilled for the banana clips. For added functionality, I added a power switch and power LED to the top of the enclosure so the power to the AR-2 board could be turned on and off. The LED lights up when the AR-2 board is turned on. With the electronics tested and enclosed, it was time to move on to the computer.

I chose Red Hat 7.3 as the operating system. A fresh install was done, choosing Server as the installation type. Once the Red Hat installation was completed, I began customizing the box. I copied ar-2.c, the hacked-up version of upscheck.c, to the box and compiled it. After it successfully compiled, I copied the resulting binary ar-2 to the /usr/bin/ directory, because the AR-2 program runs from cron at specific times.

I wanted to allow remote administration of the cron jobs that ring the bells, so I really didn't feel comfortable with them running as root. I therefore decided to add a user for the specific task of ringing and gave it the user name bell with the command, `useradd bell`. The user bell has no need to log in locally or remotely, so it is a good idea to change the user's shell to a non-functioning one with the command `usermod -s /sbin/nologin bell`.

A problem with permissions arises here that needs attention. In Red Hat 7.3, the /dev/ttyS* devices have their permissions set so that only root and members of the group uucp have read and write access to them. User bell, however, needs to be able to write to the serial ports in order to control the attached relay device. By default, the user bell does not have read or write access to the serial devices. To solve this problem, I decided to make the user bell a member of the uucp group. As root, I added the user bell to the group uucp with the command `usermod -G uucp bell`. Now bell can use AR-2 on the /dev/ttyS0 device to manipulate the control lines of the serial port.

For remote administration of the bell's cron jobs, I choose a Web-based tool with which many Linux users are

familiar, Usermin. I also downloaded Webmin for the purpose of configuring Usermin and installed them both. After installation, I logged in to Webmin and configured Usermin with the Usermin Configuration module. Because only user bell should be able to use Usermin, I used the Allowed Users and Groups module and selected "Only allow listed users"; bell, was of course, the only user I added to the list. Next, I used Available Modules to restrict all but the Scheduled Cron Jobs module. Last but not least, I went into the User Interface module and clicked Yes for the Go direct to module if user has only one option. Now bell is the only user that can log in using Usermin, and when user bell does log in, Usermin jumps straight to displaying Scheduled Cron Jobs. I also turned on mandatory SSL encryption. Webmin wasn't needed for anything else, so I stopped it with `service webmin stop`. To make sure it does not start back up on a reboot, I entered `chkconfig webmin off`.

Now we can log in to Usermin as the user bell and easily schedule the times at which the bells should ring and how long they should last by using the command `/usr/bin/ar-2 /dev/ttyS0 2 2`. Employees can feel comfortable knowing the bell always lets them know when to start running. Management can rest easy knowing they have yet another tool to help keep things running smoothly, and I can rest easy knowing that it's running on Linux, which means I don't have to worry about constantly babying it.

Further Possibilities

This was a simple exercise that illustrated the use of controlling devices using a relay over a serial interface. The low-cost relay used in this example is capable of controlling any number of electrical devices. Also available are relay boards that can be expanded to up to 128 relays per serial port, not to mention analog and digital input cards.

Jason Ellison is from Fairhope, Alabama, and currently is a network analyst for Delphi. He also maintains several Linux and AIX systems at Delphi.

RESOURCES

Electronic Energy Control, Inc.: www.eeci.com

Red Hat: www.redhat.com

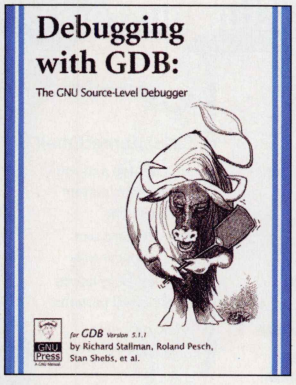
UPS HOWTO: www.tldp.org/HOWTO/UPS-HOWTO.html

Usermin and Webmin: www.usermin.com and www.webmin.com

The Definitive GCC Reference Manuals

GNU Press

GNU PRESS ▪ 617-542-5942 ▪ www.gnupress.org ▪ Free Software Foundation

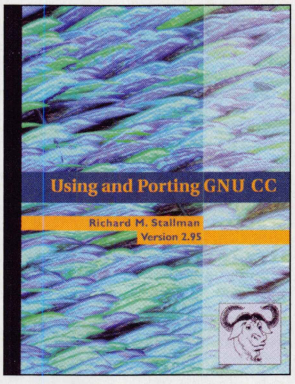


Debugging with GDB

for GDB version 5.1.1

ISBN 1-882114-88-4
346 pages • $25

Debuggers making you batty? This top selling tutorial is designed so the reader can utilize GDB after finishing the first chapter. Finish the book and you will be a master!

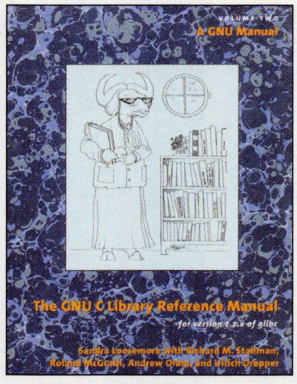


Using and Porting GNU CC

Version 2.95

ISBN 1-882114-38-8
588 pages • $35

A thorough reference guide to installing, running, debugging, configuring and porting GCC. Covers C, C++, Objective C and Fortran front-ends.

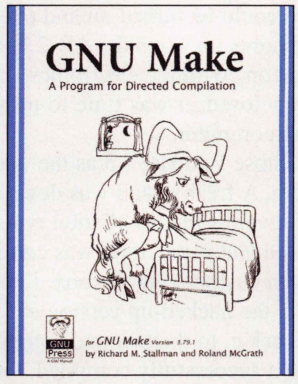


The GNU C Library Reference Manual

for version 2.2.x of glibc

ISBN 1-882114-55-8
2 vol. • 1,275 pages • $60

A comprehensive guide to implementing the standard C libraries. Contains code examples and usage recomendations. Useful for system admininstrators and programmers.



GNU Make

for GNU Make Version 3.79.1

ISBN 1-882114-82-5
196 pages • $20

This best selling tutorial is indispensible to free software system maintainers. Written by the program's authors, it also includes an introductory chapter for novice users.

Testing Applications with Xnee

Is your clicking finger sore from testing your GUI program? Script your regression tests with Xnee.

BY HENRIK SANDKLEF

Xnee can record user actions during a session and then replay those actions. By recording sessions when testing a program, Xnee automatically can test the program for you later. These test sessions can be replayed before every release, or every night, to ensure the quality of your program. Is it as easy as it sounds? Almost.

Xnee doesn't test only GUIs. You also can use Xnee to test command-line programs by making a few test scripts that test all the options of a command-line program and analyze the results. Xnee also has been used to test how much traffic is being sent over a large network with numerous thin clients. Support for distributing events to multiple displays has been added if you want to test the same cases on multiple machines at the same time. Besides testing programs, Xnee also is used to demonstrate programs. In this case, Xnee acts as a patient demonstrator, doing the same job over and over again without complaining.

The History of Xnee

In 1997, Henric Johansson and I wrote our master's thesis on recording and replaying X events. We implemented a nonfree recorder and replayer for a Swedish company for its internal purposes. After finding a job, I often lacked a free test program for X11, so I decided to implement one on my own, using the experience gained from the thesis. The Xnee Project started in the summer of 1999 and was licensed under GPL from the start. In November 2002, version 1.0 was released, and by the end of February 2003, Xnee was dubbed a GNU package.

Introduction

Before we go on with Xnee, this short introduction to X explains a lot of the terms used in this article. X is a window-based user interface system for various platforms. The X server is a program that handles all the hardware and actually does the drawing on the screen. On GNU/Linux systems, XFree86 is the most-used X server. X programs are known as clients; examples are xterm and Galeon. The clients communicate with an X server using the X protocol.

In this article we concentrate on the packets used to send information between the X server and its clients. These packets are called Event, Request, Reply and Error and are referred to here as protocol data. The following list shows the X11 protocol data description:

- Request is sent by the client to the server. The server is asked to perform some action or to send some information.
- Reply is sent by the server to the client as a response to some request from the client. Not all requests result in a reply.
- Event is sent by the server to the client to inform the client of user input or that something has happened that the client may want to do something about, for example, the client is out of focus.
- Error is sent by the server to the client if a request wasn't valid.

The most interesting thing here is every time the user interacts with the computer using the mouse or the keyboard the X server sends the appropriate client one or more events. Some of these events result directly from user input. These events are referred to as device events. The device events are ButtonPress, ButtonRelease, MotionNotify, KeyPress and KeyRelease. If we could record all of these events during a session, we would have a complete description of all the actions the user performed. If we had a robot that could read these events if they were printed to a file or on paper, the robot then could interact with the system as the user did when recording, and we would have ourselves a test robot. Or, even better, if we had support for faking those events, we would have a test robot made of software. Fortunately, support exists for both recording and replaying in X.

To record X protocol data we can use the extensions RECORD or XTrap. There are other ways to accomplish recording, such as sniffing the X socket, but we'll focus on RECORD as it's what Xnee uses. To replay events, we can use both the XTest extension and the RECORD extension. During replay, the RECORD extension is used to synchronize what's happening when replaying with what happened when the session was recorded.

The RECORD extension sends copies of the data sent between the clients and the server to the client requesting it. Using the RECORD extension, Xnee can record all protocol data the user wants and save it to a file to replay later.

The XTest extension can reproduce or fake all device events. This extension lets Xnee fake user actions, such as moving the pointer, pressing and releasing a key or pressing and releasing a button. No other data can be replayed.

Xnee Installation

Xnee is a GNU package, and the sources can be found at the Xnee home page. Download the latest source; as of this writing, the latest version is 1.0.6. Extract the package, and then configure, build and install it:

```
tar zxvf xnee-1.0.6.tar.gz
cd xnee-1.0.6
./configure
make
make install
```

Table 1. X Events at the Start of a Galeon Test

Protocol Data Name	User or Client Action
MotionNotify	The user moves the pointer to the Galeon launch icon.
ButtonPress	The user presses the button and Galeon starts.
CreateNotify	Galeon is started and the window is created.
VisibilityNotify	The start page is loaded and visible to the user.
MotionNotify	The user moves the pointer to a link on the loaded page.
ButtonPress	The user clicks on the link.
VisibilityNotify	The new page is loaded and visible to the user.

Table 2. How Xnee Replays a Test Session

Protocol Data Name	Xnee Action
MotionNotify	Xnee moves the pointer to the Galeon launch icon.
ButtonPress	Xnee presses the button and Galeon starts.
CreateNotify	Xnee waits for this event to be sent. When Xnee receives a CreateNotify notice, it continues with the next event in the file.
VisibilityNotify	Xnee waits for this event to be sent. Because the network is down and the page can't be loaded, this event isn't sent. Xnee continues to wait. Eventually the event is sent and Xnee can continue.
MotionNotify	Xnee moves the pointer to a link on the loaded page.
ButtonPress	Xnee clicks on this link.
VisibilityNotify	The new page is loaded and visible.

RPM packages are available at the home page, and Xnee also is available in the FreeBSD ports tree. Xnee comes with a user manual and a developer manual in various formats. The TeX sources to the manuals are distributed with Xnee and covered under GNU FDL. Instead of building the documents yourself, you can download the format of your choice (PDF, HTML, INFO and TXT) from the Xnee home page. As of this writing, the Xnee documentation version is 1.0.4. Extract the documents:

```
tar zxvf xnee-doc-1.0.4.tar.gz
```

When running Xnee, make sure the RECORD extension is enabled. On XFree86 make sure the RECORD module is loaded. Open the XFree86 configuration file, typically /etc/X11/XF86Config-4, and look in the Module section. The following line should be included:

```
Load  "record"
```

See the Xnee FAQ for more information about this.

Simple Usage Examples

We don't go into any details about Xnee in this section, but rather begin slowly with a simple example. Start Xnee with the --all-events option. This sets up the recording of a few events. The option is not useful when doing serious Xneeing. It is intended to simplify your introduction to using Xnee:

```
xnee --all-events
```

When moving the mouse or pressing the buttons or keys, Xnee prints information about the action. We move on to record a simple session that we replay immediately. To record 20 mouse motions, start Xnee like this:

```
xnee --record --out session1.xnr \
--device-event-range MotionNotify --loops 20
```

The options on the command line mean use recording mode (--record), save output in a file called session1.xnr (--out session1.xnr), record the device event MotionNotify (--device-event-range MotionNotify) and record 20 events (--loops 20).

To replay this event, start Xnee like this:

```
xnee --replay --file session1.xnr
```

The options on the command line mean use replay mode (--replay), and read data to replay from file session1.xnr (--file session1.xnr).

Setting Up Recording Ranges

Xnee uses ranges to explain what is to be recorded. Ranges have a start value and a stop value. The following data can be recorded: core-requests, device-event, delivered-event, error, reply, extension requests and extension replies. We don't describe the above data in this article. If you want to read more, see the RECORD extension documents. When, for example, you want to record the device-event

MotionNotify, use:

```
--device-event-range MotionNotify
```

To record the events from KeyPress up to MotionNotify and CreateNotify, use:

```
--device-event-range KeyPress-MotionNotify,\
CreateNotify
```

You can use the number corresponding to the event name instead of the name itself if you want shorter command lines. To find the number for the data you want to record, use Xnee's --print-data-name option:

```
xnee --print-data-name
```

Stopping Xnee

You can stop recording by setting the number of the data to record (--loops option), or you can interrupt Xnee by sending a TERM signal (pressing Ctrl-C in the terminal window where you started Xnee). Alternatively, you can dedicate a modifier and key combination that won't be used to do anything else during recording. Setting the modifier and key is done with the --stop-key option. To set up Xnee so it stops recording when Ctrl-Alt-A are pressed, add the following to the command-line option:

```
--stop-key Control+Alt,a
```

Synchronize

But why even bother to record data other than device events when you can't replay it? Xnee uses that other data to synchronize, which is where things get complicated. Think of recording a session when using Galeon or any other Web browser. When recording, everything goes well and the network is up and running. But when replaying the Galeon session, you can't reach the Internet. If not for synchronization, Xnee might replay user events such as clicking on a link on the Web page. If Galeon could not load the page, it is not useful to continue the replay until the network is up and the page can be loaded.

When recording other data, we can use it to synchronize the session. For example, if we record the data sent when displaying the Web page in the Galeon window, we can wait for the same data to be sent when replaying. This ensures that the Web page is loaded before we go on and replay the coming events. In this example, we skip a lot of the X protocol data sent when recording in order to keep it simple (see Table 1). When replaying this simple session, Xnee uses the same events (see Table 2).

What to Synchronize

Although synchronization is needed, finding the right data to use for synchronization may be difficult. Xnee solves this with plugin files that specify what should be recorded for a range of

See the Future · Shape the Future

The 3rd Annual
O'REILLY®
EMERGING TECHNOLOGY CONFERENCE™
February 9–12, 2004
Westin Horton Plaza, San Diego, CA

Save up to $430! · Register by January 9, 2004
Register today at conferences.oreilly.com/etech

Tracks include:
Interfaces & Services
Untethered
Social Software
Location
Hardware
Business Models

Featured Speakers:
Marc Smith Microsoft Research
Helen Greiner iRobot Corporation
Pertti Korhonen Nokia
Tim O'Reilly O'Reilly & Associates
Sam Ruby IBM
Eric Bonabeau Icosystem Corporation
Russell Beattie Independent Technical Consultant
Andrew Huang Xenatera LLC
Danny O'Brien Need To Know
Michele Chang Intel Research

©2003 O'Reilly & Associates, Inc. O'Reilly is a registered trademark of O'Reilly & Associates Inc. All other trademarks are property of their respective owners.

applications. These plugins are named after the applications they are intended to test. If you want to test a browser you've written, it would be a good idea to use the Galeon plugin. Sometimes, though, no plugins are suitable for your program, and you need to find the right protocol data to synchronize. The following example hopefully makes it easier for you in the future. We chose gnumeric as a program for which the right options need to be found. First, launch gnumeric. Then start Xnee in a terminal emulator with the following options:

```
xnee  --delivered-event-range \
EnterNotify-MappingNotify --human-printout \
--loops 1000
```

This generates a lot of useless events that fill the screen, so stop Xnee. Filter out those useless events by excluding them when setting ranges:

```
xnee --delivered-event-range \
EnterNotify-KeymapNotify,VisibilityNotify- \
MappingNotify --human-printout  --loops -1
```

This looks better. Now, start recording a session with Xnee with the following options:

```
xnee  --delivered-event-range \
EnterNotify-KeymapNotify, \
VisibilityNotify-CirculateRequest, \
SelectionClear-MappingNotify --loops \
1000 --out session1.xnr
```

Type some stuff into the gnumeric spreadsheet and use the menus to insert today's date or other input. When you're done, go to the terminal and press Ctrl-C to stop recording. It is now time to replay your session. Set gnumeric in the same state it was in when you recorded. Launch Xnee in replay mode like this:

```
xnee  --replay -f session1.xnr
```

Xnee sometimes pauses when replaying the session. This happens if the protocol data isn't sent in the same order as it was recorded. Xnee pauses execution for a while in order to wait for the expected data (as read from file) to be sent by the server. Eventually, a timeout expires and Xnee tries to continue. If Xnee can't synchronize between the recorded data traffic and the data traffic as sent when replaying, it bails out.

Xnee supports giving record options through plugins. When you have found the settings for your applications, save them in a plugin file. The syntax of a plugin file is similar to the command-line options. The easiest way to crate a new plugin is to copy an old one, fill in your settings and then rename it to some appropriate name. Xnee is distributed with plugins for different clients. If you want to send a plugin file for your application to Xnee, please do. The Xnee home page has instructions for how to contribute.

If you have a program that creates windows for user feedback, you have to make sure these windows pop up at the same location. Xnee records all device events with coordinates referring to the root window, not the window that was created.

To ease recording, make scripts that start Xnee with the right settings for a specific purpose. You can add a launcher to your panel or add a menu item to your window manager menu.

Conclusion

Xnee has seen a lot of activity lately, mainly due to feedback from Xnee users. We hope you consider Xnee for your project. Happy testing and happy hacking.

Henrik Sandklef lives in Gothenburg (Sweden) with his wife and daughters. He spends most of his time awake with his family, cooking, hacking and evaluating GNU software, and occasionally, he tries to play football. You can reach him at hesa@gnu.org.

RESOURCES

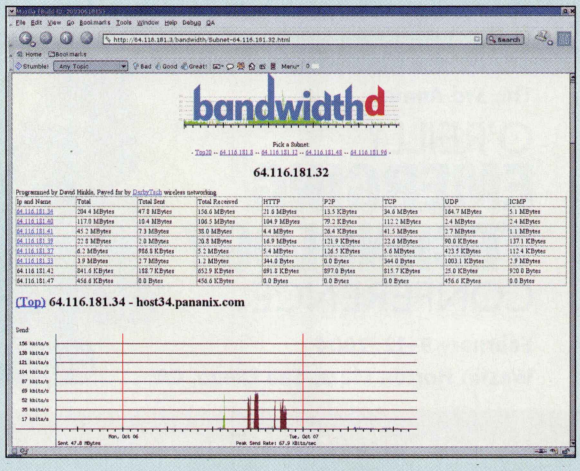
GNU: www.gnu.org

Kenton Lee: www.rahul.net/kenton/xsites.html

XFree86: www.xfree86.org

Xnee: www.gnu.org/software/xnee/www/index.html

bandwidthd: bandwidthd.sourceforge.net



The one reason you might want to consider bandwidthd over MRTG or Cacti is that bandwidthd graphs individual IP addresses instead of capturing raw traffic volume. bandwidthd also attempts to categorize the traffic into TCP, UDP, ICMP, VPN, HTTP and P2P traffic per host. That is, you can see what each host is doing with the bandwidth it's using. Perpetual bandwidth hogs and even sudden virus infestations can be seen on the graphs once you have a baseline. Requires: libgd, libpng12, libpcap, glibc, libXpm, libX11, libjpeg, libfreetype, libz, libm and libdl.

—DAVID A. BANDEL

Intermediate Emacs Hacking

Turn Emacs into your editor of choice with this easy customization tool. **BY CHARLES CURLEY**

Customizing Emacs is important to get the most out of it. You can change the way it operates to reflect your way of doing things, which in turn makes you more efficient. Just as custom-built boots fit better than factory boots, a customized Emacs fits you better than standard, off-the-shelf Emacs.

Emacs changes can be session-specific or permanent. You can customize Emacs directly by executing commands in the mini-buffer or by modifying variables using the set-variable command. These changes are volatile, meaning you lose them when you end your session. To make permanent changes, you can create or modify an init file. Emacs examines several init files when it loads. Probably the easiest way to customize is to edit .emacs, stored in your home directory. Before loading your .emacs file, Emacs loads default.el from your library path. Emacs also looks in its load path for site-start.el, which system administrators may use to provide site-wide customization. Alternatively, you can make permanent Emacs changes through the Customize menu (Options→Customize Emacs in version 21.1 and up). This gives you a GUI-based front end for modifying your .emacs file.

Using Customize

Like most GUI front ends, Customize is not as powerful as hacking the text configuration file, but it is easier to use. Fire up the Customization group, and you notice that Emacs builds the menus on the fly. That way it's always up to date (Figure 1).

To navigate Customize's tree structure, either point to a button and press Return or click on the button. Full Emacs searching also works in the Customize buffers. Each level in the tree is represented in a buffer, and you can manipulate Customize buffers as usual. For example, when you are done with a level, kill the buffer.

I'd like to turn on the PHP speedbar whenever I enter PHP mode, for example, when I visit a PHP file. To do this, I follow the menu tree to PHP mode customization (Figure 2). As you can see, I've toggled the state but haven't yet saved my changes. I can set it only for this session, or I can save the change for future sessions. When I do the latter, Emacs edits my .emacs file. You can verify the change by searching for the variable name in .emacs.

If you aren't sure of the name of a variable or where to find it in Customize's vast tree, you can use regular expression searches on variable names and their contents. If you want to change how Emacs prints, you could search on the regex

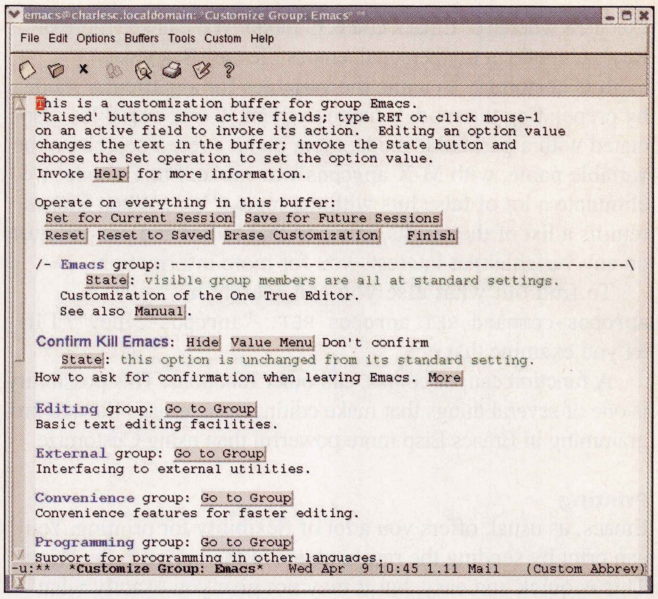


Figure 1. The Top-Level Customization Menu

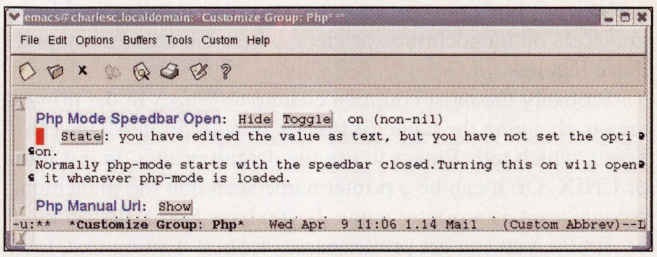


Figure 2. Setting the Speedbar to Turn On in PHP Mode

"print" and Emacs would build a custom Customize menu for you. See the menu under Options→Customize Emacs for these and other options.

Editing Your .emacs File

You also can modify Emacs' behavior by editing ~/.emacs directly. This is a good way to add function and insert bulk customizations you may learn about from other people. For example, C mode can be modified by setting variables that affect how it operates.

Many modes are customized on a per-buffer basis. This means you write a short function that sets the variables and set that function to be executed whenever Emacs enters the mode. The outline of it is:

```
(defun rays_c_mode ()
  "ray's c/c++ mode hook"
  (message "Loading Ray's C mode...")
  ...
  (message "Loading Ray's C mode... Done")
  )
(add-hook 'c-mode-common-hook 'rays_c_mode)
```

The Lisp function defun defines a function, in this case rays_C_mode. The function takes no parameters; it prints out messages only for the user's benefit. The last line adds the function rays_c_mode to C mode's mode hook, that is, a list of functions

executed whenever Emacs enters C mode. You can see more of Ray's C mode in my personal .emacs file (see Resources).

It is customary to name the variables for a particular mode by prepending the mode's name to them. To see variables associated with a particular mode, then, make a regex search on the variable name, with M-X `apropos-variable`. For C mode, we eliminate a lot of false hits with the regex "^c-". When Emacs returns a list of the results, move to that buffer and press Return on any variable that interests you for more information.

To find out what else you can search on, try M-X `apropos-command RET apropos RET.` "apropos-zippy"? I'll let you examine that one.

A function can, of course, call other functions. This possibility is one of several things that make editing your .emacs file and programming in Emacs Lisp more powerful than using Customize.

Printing

Emacs, as usual, offers you a lot of flexibility for printing. You can print by sending the raw contents of the buffer to the printer. This is quick and easy, but it may not give you exactly what you wanted. You have far more control over PostScript printing, so that may be the way to go. For one thing, font-locked (colorized) buffers print in color on color printers and in grayscale on monochrome printers, where Ghostscript supports those features.

Probably the most common change to Emacs in the printing area is changing the name of Emacs' default printer. This can be nil, which tells Emacs to use the default printer, lp in Linux or UNIX. Or, it can be a printer name such that the lp dæmon recognizes it as a printer name. If you have Red Hat's printtool or similar, you can get printer names from it. Failing that, look in /var/spool/lpd/ for the names of your printers. Emacs can use any printer for which you can define a local queue, including remote printers.

Two variables are set to indicate the printer, one for PostScript printing and one for non-PostScript printing. The code below shows how I set up Emacs under Linux and Windows. The Windows definition uses a remote printer on the computer charlesc. Setting both printers to the same computer works only because the server is a Linux box. The printcap detects PostScript and runs it through Ghostscript before printing it:

```
;; Begin setup for printing on Win32
(if (and  (>= sams-Gnu-Emacs-p 20)
          (memq window-system '(win32 w32)))
    (progn (setq printer-name "//charlesc/lp")
           (setq ps-printer-name "//charlesc/lp"))
  )
;; End setup for printing on Win32

;; Begin setup for printing on Linux
(if (and  (>= sams-Gnu-Emacs-p 20)
          (string-equal system-name
                        "charlesc.localdomain"))
    (progn (setq printer-name "lp")
           (setq ps-printer-name "lp"))
  )
;; End setup for printing on Linux
```

Emacs' PostScript printing is very powerful. By default, it prints a gray box at the top of each page with the buffer's name, the data, a page number and a count.

You can set a text string and other characterstics of a watermark on each page or on selected pages. For example, a watermark of Preliminary or Draft is a good idea for code reviews; see the variable ps-print-background-text. You also can use an EPS image, such as a picture of Tux or the Free Software Foundation's GNU logo, for a watermark.

We geriatric cases who don't like 8.5 point type can change the value of ps-font-size. The value contans two numbers, the first for landscape and the second for portrait printing. Whether to print landscape or portrait is controlled by the variable ps-landscape-mode.

You can modify the default PostScript header and add a footer as well. For two-sided printing, you may specify left and right headers and footers. And if you want to save trees, look at ps-n-up-printing. It lets you print multiple pages on a sheet of paper.

Font Selection

Emacs supports multiple frames or windows. You can launch another frame with Ctrl-X 5 2, or remove one with Ctrl-X 5 0. The initial frame created when Emacs is launched has a number of graphics characteristics defined by the variable initial-frame-alist. Subsequent frames are governed by default-frame-alist. Use these two variables identically. Each variable is a list of sub-variables and their values, rather like a hash in Perl. For example, to set the initial position of the first frame on the screen, use:

```
(setq initial-frame-alist
     '((top . 40) (left . -15)
       (width . 96) (height . 40)
       (background-color . "Gray94")
       (foreground-color . "Black")
       (cursor-color     . "red3")
       (user-position t)
       ))
```

This definition sets the Emacs initial frame 40 pixels from the top of the screen, 15 pixels from the right (hence the negative number for left), with a width of 96 characters and a height of 40 lines. It sets the default background and foreground text colors and then sets the cursor color.

This definition is also where you set your font, if you don't like the default. The program xfontsel comes with XFree86, and you can use it to find a suitable font (Figure 3). Press Select as an option, and xfontsel puts it into the clipboard. Add another pair of parentheses to the definition of initial-frame-alist, insert the phrase `font .` and insert a pair of quote marks. Then recover the font definition between the quotes with Ctrl-Y, like any other clipboard entry:

```
(font . "-adobe-courier-*-r-*-*-*-140-*-*-*-*-*-*")
```

If you are wondering how I get the colors in my screenshots, that's how I do it.

If you have problems with these settings, look in your .Xresources file for any Emacs settings. Any settings in .Xresources overrides these definitions. To get rid of the over-

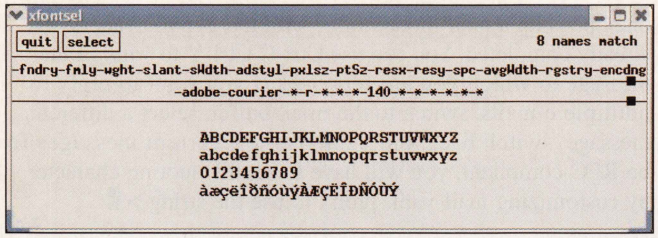


Figure 3. Selecting a Font with X's Utility Program xfontsel

rides, comment them out of the .Xresources file and restart X.

Until recently, Emacs had no support for variable-width fonts, and fixed-width fonts are fine for most purposes. But with support for proportional fonts and non-English character sets included in recent versions, you now can define your own sets of fonts within Emacs. Fontsets allow word-processor ease of control over fonts. For examples, take a look at the headers in Emacs' Info and the Customize menus; see also Emacs' info nodes on fontsets.

For more information on fonts under X, see Emacs' info node on Font Specification Options.

The Speedbar

I explained how to automate turning on the speedbar in the discussion of the Customize Emacs system. Speedbar is a separate frame (window) that allows mouse-click navigation among Emacs buffers. As you can see from the illustration, the speedbar allows for tree structures, like Emacs' Info system. Click on the + to open subnodes, and click on the – to close them (Figure 4).

When you are editing in certain modes—Rmail, Info and GUD, for example—the speedbar shows other selections to be edited in that mode. For example, when you are in Info mode, the speedbar displays nodes. Otherwise, it shows files in the directory where the file in the current buffer is located.

If you have a lot of files open in Emacs, the speedbar is a useful tool. I often have more than 30 files open simultaneously, and the speedbar helps me manage and switch between them.

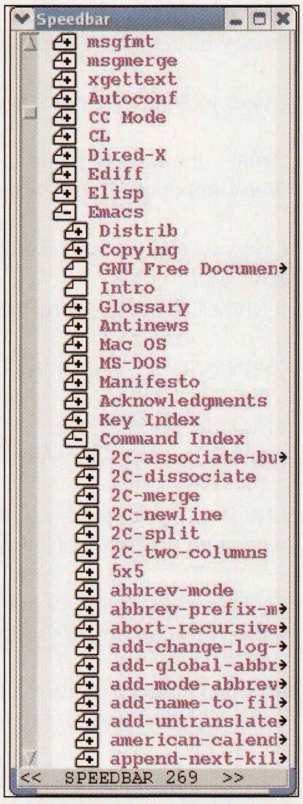


Figure 4. Emacs' Speedbar Point-and-Click Browsing Interface

Emacs Is for E-mail

In the article "Getting Started with Emacs", *LJ*, March, 2003, I demonstrated how to use Emacs as a server, letting programs like crontab and mutt use Emacs for file editing. To carry this further, you can use Emacs as the editor for any application that can call on an external editor, including mail readers. However, Emacs has at least two mail modes and a powerful newsreader called GNUS.

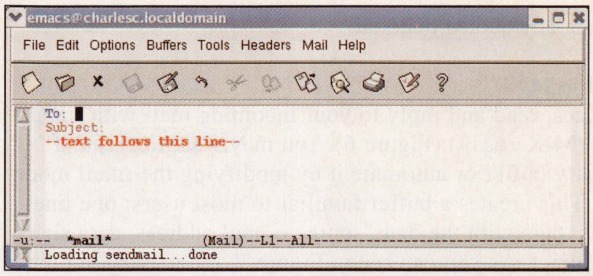


Figure 5. A Skeleton E-mail in Emacs' Mail Mode

To send a message, Ctrl-X M (or M-X `mail`) puts you in mail mode. Simply edit a message and send it (Figure 5). As the screen capture shows, you see a skeleton of an e-mail ready for you to fill in the blanks. Use control character sequences to move to (and create) additional headers, like FCC. Use tab completion in the headers. There, it looks at local system users and the contents of any e-mail aliases you have defined in your .emacs.

You can insert your signature with Ctrl-C Ctrl-W, or you can have Emacs do that for you by setting mail-signature to `t` in your .emacs file. If you want to get fancy and write a Lisp program to select a signature for you based on, well, whatever you can write Lisp code to detect.

You can run a spell-checker on your message. Entering M-X `ispell-message` checks only the body of the message,

LINUX CERTIFIED
LinuxCertified, Inc.
LC2000 Series Linux Laptops
Turbo Power - Fully Configured - Amazing ROI
Linux Laptop
Open Source Training, Services and Products
1877 800 6873
www.linuxcertified.com

skipping any quoted material.

Reading Mail

In Emacs, read and reply to your incoming mail with rmail mode (M-X `rmail`) (Figure 6). You may want to create a summary buffer or automate it by modifying the rmail mode hook. This creates a buffer familiar to most users: one line per message with the date, source e-mail address, data size and subject. As you can see in Figure 7, the message in the rmail buffer is highlighted in green. The normal Emacs navigation keys work in the summary buffer.

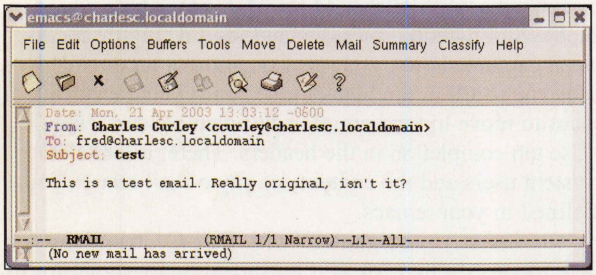


Figure 6. An E-mail Message in rmail Mode

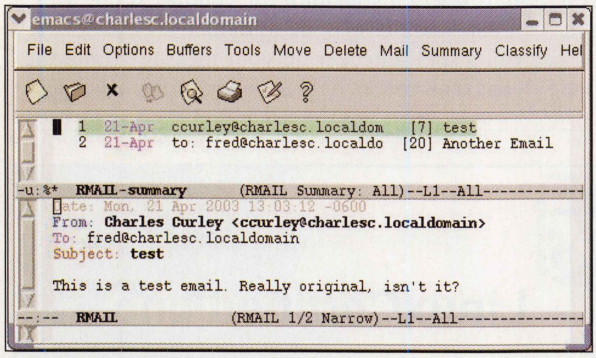


Figure 7. rmail Mode with a Summary

Emacs mail operates in a somewhat convoluted way in order to accommodate multiple operating systems. When you start an rmail buffer, it moves mail from your inbox file, typically in /var/spool/mail on Linux, into a file, ~/RMAIL. This is the file you normally edit. You can put e-mail into ~/RMAIL at any time with the G key. If you have a POP or IMAP account, try using fetchmail to put your mail in the inbox.

Emacs uses the Babyl mail file format. You can export individual messages as text files; entire rmail files can be exported in mailbox format.

Most mail readers use multiple mail files (directories, if they use maildir mail format). Emacs can shift from one rmail file to another, but you may not need to. Instead, you can create customized summaries using regular expressions and other search patterns. You can specify summaries based on recipients, a regular expression search within the subject or labels.

You can have multiple rmail files and associate each one with one or more inboxes. This means that folks with spam filters, such as SpamAssassin, already running or using procmail recipes to deliver their e-mail to separate files need not abandon that investment. Each time you visit an rmail file, Emacs gets any new mail from the associated input files.

You can reply and forward e-mail in rmail mode. Either one opens up a mail mode buffer with the e-mail headers already completed. You can use Ctrl-C Ctrl-Y to yank in the message to which you are replying. If you want to reply to multiple e-mails, switch to the rmail buffer, select a different message, switch back and yank the new current message. To be RFC-compliant, you will have to set the quoting character by customizing mail-yank-prefix to use the string >.

Charles Curley (**www.charlescurley.com**) teaches Linux at two Wyoming colleges. He also writes software and articles and books, using open-source software tools such as Emacs.

RESOURCES

Author's .emacs:
www.charlescurley.com/~ccurley/emacs.init.html

Emacs Beginner's HOWTO: **www.tldp.org/HOWTO/Emacs-Beginner-HOWTO.html**, or possibly already on your computer with the rest of the LDP docs.

Emacs' Built-in Help System: Ctrl-H

Emacs for Vi Users: **grok2.tripod.com**

"Emacs: the Free Software IDE", by Charles Curley, *LJ* June 2002:
www.linuxjournal.com/article/5765

Emacs Wiki: **www.emacswiki.org/cgi-bin/wiki.pl**

fetchmail: **www.catb.org/~esr/fetchmail**

"Getting Started with Emacs", by Charles Curley, *LJ*, March 2003

GNU Emacs Home Page:
www.fsf.org/software/emacs/emacs.html

GNU Emacs Lisp Reference Manual:
www.gnu.org/manual/elisp-manual-20-2.5/elisp.html

GNU Emacs Tutorial (Old, but Still Useful):
www.futureone.com/~sponge/tutorial/emacs/index.html

How do I make common modifications to my Gnu Emacs .emacs file? **www.yak.net/fqa/124.html**

Procmail: **www.procmail.org**

Programming in Emacs Lisp: An Introduction, by Robert J. Chassell:
www.gnu.org/manual/emacs-lisp-intro/emacs-lisp-intro.html

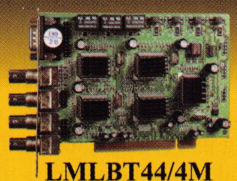
SpamAssassin: **spamassassin.org**

Tips—emacs "Nifty ways to get your work done in /the/ editor":
www.portico.org/index.php3?catList=28

"very unofficial .emacs home", by Ingo Koch:
www.dotemacs.de

Linux Media Labs

LMLM4
MPEG-4 Hardware Audio/Video Encoder
RTSP/RTP/RTCP live streaming
1.5 Mbit/sec for full rate video
720x480 (576) @ 30 (25) fps
Supported by MPlayer, DivX



LMLBT44/4M
Multichannel Video IO
4/8 Channels Video Capture : 720x480
4 Input Sensors/ 3 Output Alarms
Fully Supported by Linux Kernel

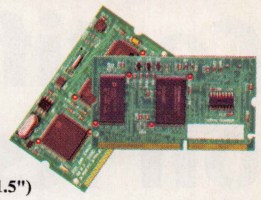
LINUX MEDIA LABS

www.linuxmedialabs.com
e-mail : sales@linuxmedialabs.com
1-888-795-4735 or 1-719-231-3173
Free Gift with any purchase : Promotion Code #200305LJ

TINI® GIANT Internet Appliance Engine

SoM-400EM

- DS80C400 TINI® CPU
- IPv6 Fast Ethernet
- 3 Serial Ports & SPI
- CAN 2.0B & 1-Wire® Ports
- Programmable in Java™ or C
- 2 MB Flash & up to 2 MB RAM
- Telnet, FTP, and HTTP Servers
- Typical Power Consumption < 1 Watt
- Robust, Free Java™ Development Tools
- Real Time Clock & Nonvolatile Memory
- Small, 144 pin SODIMM form factor (2.66" x 1.5")



The SoM-400EM is a System on a Module, based on the Dallas® Tiny InterNet Interface (TINI®) Processor. This 8051 code-compatible processor makes it extremely easy to create a smart Network / Internet capable device and comes complete with Embedded Java™, Unix-Like OS and File System. Write sophisticated network applications in days instead of months. Unit pricing starts at $105. Optional Carrier/Socket board, Enclosure, and Power Supply available.

EMAC, inc.
E QUIPMENT M ONITOR A ND C ONTROL
Phone: (618) 529-4525
Fax: (618) 457-0110
Web: www.emacinc.com
Since 1985 OVER 18 YEARS OF SINGLE BOARD SOLUTIONS

SPEEDY PUPPY

www.speedypuppy.net

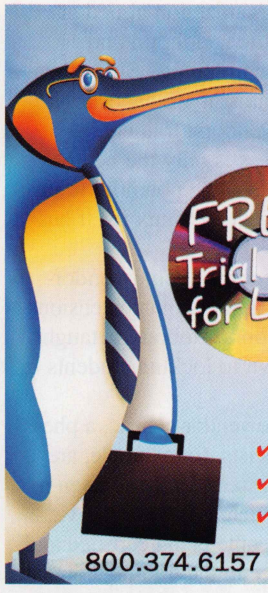
Fast & Reliable Web Site Hosting

Unix-based hosting with:

- PHP4, Perl, Python
- MySQL databases
- Shopping cart package
- **SECURE** POP/IMAP/ Webmail email

$8.95 per month

Instant activation with no setup fee!
www.speedypuppy.net



Fitrix

Exact Fit Accounting & Business Software

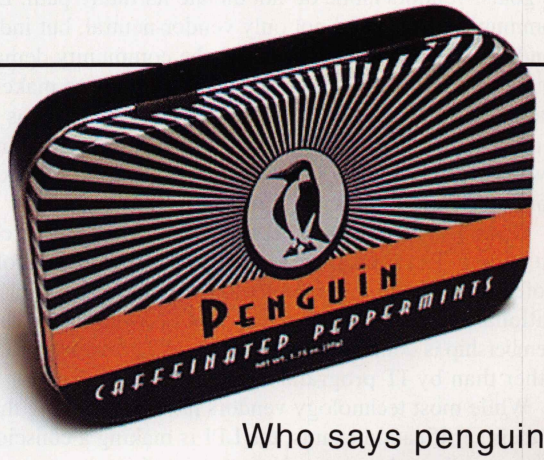
FREE Trial CD for Linux!

Accounting software that allows for continual improvement!

- ✔ Integrated business & accounting software suite
- ✔ Designed for easy modification with no limits
- ✔ High-end technology at an affordable price!

800.374.6157 • 770.432.7623 • www.fitrix.com

www.linuxjournal.com



Who says penguins can't fly?

http://store.linuxjournal.com/

Turning IT Certification on Its Ear

How to offer better certification, for more people, at a lower price, without vendor lock-in.

BY EVAN LEIBOVITCH

The Linux Professional Institute (LPI) was started in 1998 as a Linux certification organization aimed at removing obstacles to open-source adoption and increasing the skill level of its community. Since then, LPI has delivered more than 35,000 exams in almost every country and has gained the support of vendors, trainers, employers and governments. This success is only the beginning, however. As an organization, LPI has the potential to turn the world of IT certification on its ear, just as Linux itself has disrupted the software world.

Technology-related training has become such a vendor-driven commodity that it's merely nothing more than extensions of marketing programs of software vendors. Nobody is taught technique anymore; it all has come down to locking students in to one tool.

IT certification now is expected to be vendor-centric, a phenomenon unique to the IT field. Pharmacists, for example, are not trained to understand only one brand name of drug. Even in the case of "vendor-neutral" programs, such as CompTIA, vendors still control the certification. The people being certified and those who would hire them have no voice in the training content.

Although LPI has accepted vendor sponsorships to further its goals, vendors alone do not dictate its future path. LPI is community-driven and not only vendor-neutral, but indeed vendor-independent. For instance, the community demanded desktop Linux certification, so LPI is working to make it a reality. More significantly, what we are trying to do as a body is bring the IT certification world kicking and screaming toward the peer-review model that people have come to expect from other professions.

Given its perspective, LPI finds itself having less in common with other IT certification programs available to technology professionals and more in common with groups such as the National Organization of Competency Assurance, whose membership is dominated by medical and vocational certifications rather than by IT programs.

While most technology vendors have been hiking the costs of their certification exams, the LPI is making a conscious effort to reduce the costs of its exams and increase accessibility in every country. In some countries, the $100 US charge for an LPI exam is an IT worker's salary for a month. Since LPI's goal is to advance open source rather than maximize revenue, we clearly have a challenge ahead. Even in rich countries, the fees for most certifications often are beyond the reach of the unemployed looking for new career paths.

A major initiative within LPI is research and development in ways to lower the cost of computer-delivered education, while improving the testing technology to deliver hands-on testing in a scalable, cost-effective manner. Our growth is dependent on the progress of Linux and open source, which has made some significant inroads in the enterprise and public sector. The City of Munich's decision in May 2003 to select Linux not only for servers but also for desktops makes it the poster child for Linux clients. Munich's decision wasn't about price; Microsoft fought hard to keep the business, according to published reports, and was priced less than the Linux option. But Linux won the bid.

Although there may be no doubt Linux has global presence, LPI faces challenges when it comes to making sure that open-source education and certification are relevant and useful in many languages, cultures and governmental structures. LPI from its central office alone can't make Linux relevant everywhere around the world. Nor would a conventional branch-office approach work either. Our community-based affiliate network is now operating in the United States, Japan, Brazil, Canada, Germany and Austria. More national affiliates are coming on-line every day; some of the newest additions include Australia, Nigeria, Bulgaria and China.

In particular, we must make the human resources community aware that Linux certification is available.

The short-term goal is to expand this network of affiliates, but we have a long-term vision as well: to make sure certification is accessible and of the highest quality possible, while being relevant in every locale. Even with our network growing by leaps and bounds, hurdles remain. In particular, we must make the human resources community aware that Linux certification is available. Organizations as a whole need to know that if they select Linux as a strategic technology, expertise is available to them in the form of trained, certified professionals who have passed the LPI program.

And while having one's training nullified because of a product upgrade is frustrating, LPI also understands that re-certification over the years is as important for the IT profession as it is any other profession. Where LPI differs from other IT certifications is that if we introduce a re-certification program, it will be based on fixed-year intervals rather than on product upgrades.

Like Linux itself, the certification program developed by LPI demonstrates the power of community involvement. Unlike any other mainstream certification program, LPI combines the strength of a worldwide grassroots network together with experts in exam development, psychometrics and delivery. As LPI moves forward, we intend to keep setting the example that all other IT certifications eventually will be compelled to follow.

Evan Leibovitch is President of the Linux Professional Institute (www.lpi.org).

Technology you can count on

64-bit!

Itanium®2 and Opteron™ Clusters

Robust Solutions for Demanding HPC Applications

Itanium2, Opteron, Xeon, Athlon and Alpha Clusters

Expert Integration...

Microway offers innovative, competitively priced, custom designed clusters with NodeWatch™ management and monitoring tools. Our solutions incorporate the latest processors, proprietary cooling and storage solutions, plus high-speed Myrinet and InfiniBand interconnects for both fine and coarse grain parallel applications.

Superior Service and Tech Support...

We understand that on-time delivery, out-of-the-box reliability and excellent ongoing technical support are critical to our users. Microway offers professional services from specialists with a wide range of expertise in HPC applications. On-site installations and training are also available.

Satisfied Customers...

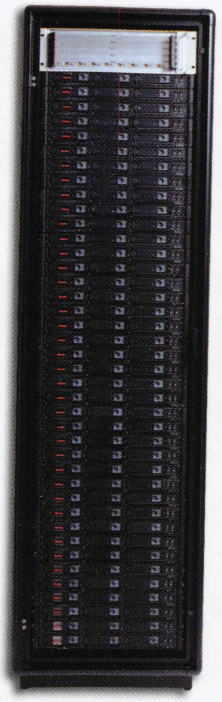
AT&T, Cessna, GE, GSK, Johnson & Johnson, LANL, LLNL, MBL, Millennium Pharmaceuticals, NIH, Northrop Grumman, Raytheon, Sandia, Seagate, US Air Force, Army, Navy, NASA, NOAA and hundreds of leading universities are among our satisfied customers since 1982.

"The Brain Imaging Research Center (a joint center of Carnegie Mellon University and University of Pittsburgh) decided to purchase our Linux cluster from Microway because of the proven performance of their clusters at Carnegie Mellon in the processing of high-volume brain imaging data. Microway was flexible and helpful at all stages, starting from the initial custom configuration and ending with timely delivery and full installation."

– Marcel Just, Co-Director, Brain Imaging Research Center

Call us first at ***508-746-7341***
for quotations and benchmarking services.
Find technical information, testimonials, and online newsletter at ***www.microway.com.***

See us at **SC2003**, Booth 2227
Phoenix – November 17-20



CoolRak™ Cabinet Features 3 10" 535 CFM Fans, yielding 7 degrees cooler ambient per node. Available only on Microway 1U Opteron, Athlon and Xeon chassis.

Microway 1U CoolRak™ features dual 2.8GHz Xeons with proprietary Across the Board™ Cooling

Innovative Technology and Services at Competitive Prices

- **Xeon XBlade™ Technology by Microway** Incorporates two dual Xeon XBlades in our 1U chassis for 880 Gigaflops performance per 44U cabinet. Each Xeon XBlade provides two 133 MHz PCI-X sockets enabling next generation interconnects including InfiniBand.
- **NodeWatch™** provides automatic remote monitoring of vital cluster parameters and failsafe shutdown of the cluster. NodeWatch monitors temperatures, voltages, and chassis fans, runs off the master, and can be controlled by a secure web-based GUI. Available on Microway 1U Opteron, Xeon and Athlon chassis.
- **Integrated solutions powered by Platform Computing** Workload Management and Grid Computing software solutions to optimize computing resources. From Microway's Clusterware-powered CW Series, to integration of LSF 5 on any Microway cluster, all Platform solutions are "grid ready" for future system enhancements.
- **Integrated professional MPI/Pro®** enhances cluster value with middleware for high performance, scalable and robust message passing.
- **Fully redundant, highly-available storage systems** based on 2Gbit fiber channel technology for multi-terabyte storage requirements. State-of-the-art storage directors for full access to storage from any cluster node. Easily scales to address rapidly expanding storage requirements.
- **Consulting Services:** Microway offers expertise in designing custom software for financial analysis, genomics, proteomics, plus data warehousing, project management tools, network/data center design, and lab integration/management systems.

Microway®
Technology you can count on℠

AMD Opteron™

intel inside ITANIUM 2

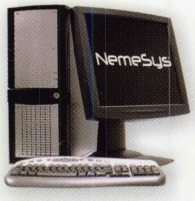
intel inside XEON

Experience the most computing power in the smallest footprint possible...

AMD Opteron™

The RackSaver® BladeRack™, powered by AMD Opteron™ processors, is designed to support a wide variety of enterprise solutions...including yours.

Over the past 10 years, RackSaver has provided customers in every industry from Entertainment to Oil and Gas with the most powerful, reliable, and scalable high performance clusters available on the market today. Our customers understand that by increasing density in their server room they improve productivity while lowering their overall total cost of ownership. In addition, RackSaver offers complete onsite enterprise-level support 24 hours a day, 7 days a week to maximize your investment protection. Discover why industry leaders like Industrial Light and Magic, NVIDIA, Shell, WesternGeco, and Novartis rely on RackSaver for all of their high performance computing needs.



THE NEMESYS

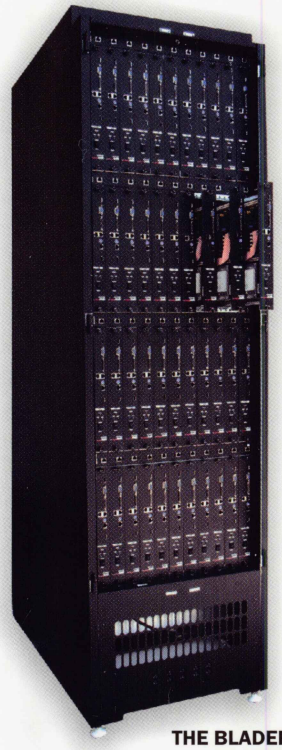
Did we mention that we sell high-end customizable graphics workstations too?

RackSaver has launched a new line of high-end graphic workstations. NemeSys™, powered by the NVIDIA Quadro® FX family, takes the leading design and digital content creation applications to a new level of interactivity. For the first time, styling and production rendering become integral functions of the design workflow, shortening the production process and enabling faster time to market.

THE HIVE CLUSTER

Need a blade solution that can fit under your desk at work and provide more power than ever before?

The HIVE™ houses up to 11 blades that can accommodate a variety of options including high-end servers, SAN, NAS-based storage, along with InfiniBand®. Both the 8 blade and 11 blade versions are fully compatible with RackSaver's RS-1100VM BladeRack line of products. The HIVE's innovative cooling technologies are specifically designed for the needs of small rack systems and also make them suitable for a wide range of environments including military, industrial or a simple office setting.



THE BLADERACK RS-1100VM

RackSaver®
Leader in High-Density Servers!

Start configuring your company's high performance supercomputing solution now by visiting RackSaver online at www.racksaver.com or call one of our Account Managers at (888) 942-3800.

Copyright ©2003, RackSaver Inc. All Rights Reserved. RackSaver, the RackSaver logo, BladeRack, and NemeSys are trademarks or registered trademarks of RackSaver Inc. Intel and Xeon and combinations thereof are trademarks of Intel Corporation or its subsidiaries in the United States or other countries.